P.S.I.

PAR
JACQUES
DECONCHAT

P.S.I.

102 PROGRAMMES POUR AMSTRAD

CONNAISSEZ-VOUS TOUTE LA COLLECTION AMSTRAD CHEZ P.S.I. ?

Pour les Amstrad CPC 464, 664 et 6128

Initiation :

— La découverte de l'Amstrad - Daniel-Jean David
— Exercices en BASIC pour Amstrad - Maurice Charbit

Programmation BASIC :

— Super jeux pour Amstrad - Jean-François Sehan
— Amstrad en famille - Jean-François Sehan
— Super générateur de caractères sur Amstrad - Jean-François Sehan
— Photographie sur Amstrad et Apple II - Pierrick Moigneau et Xavier de la Tullaye
— Amstrad en musique - Daniel Lemahieu

Maîtrise du BASIC :

— BASIC Amstrad - 1. Méthodes pratiques - Jacques Boisgontier et Bruno Césard
— BASIC Amstrad - 2. Programmes et fichiers - Jacques Boisgontier
— BASIC Plus, 80 routines sur Amstrad - Michel Martin
— Périphériques et fichiers sur Amstrad - Daniel-Jean David

Assembleur et Pascal :

— Assembleur de l'Amstrad - Marcel Henrot
— Turbo Pascal sur Amstrad - Pierre Brandeis et Frédéric Blanc

Système :

— Clefs pour Amstrad - 1. Système de base - Daniel Martin
— CP/M Plus sur Amstrad 6128 et 8256 - Yvon Dargery
— Clefs pour Amstrad - 2. Système disque - Daniel Martin et Philippe Jadoul

A paraître :

— Intelligence artificielle : langage et formes sur Amstrad - Thierry Lévy-Abégnolli et Olivier Magnan
— Animation graphique sur Amstrad - Gilles Fouchard et Jean-Yves Corre
— Simulation et intelligence artificielle sur Amstrad - René Descamps
— Le livre de l'Amstrad 8256 - Eric Baumarti

Pour tout problème rencontré dans les ouvrages P.S.I. vous pouvez nous contacter au numéro ci-dessous :

Numéro Vert/Appel Gratuit en France

05 21 22 01

(Composer tous les chiffres, même en région parisienne)

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1[er] de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

© Éditions du P.S.I. - B.P. 86 - 77402 Lagny/Marne cedex
1986
ISBN 2-86595-222-3

JACQUES
DECONCHAT

102 PROGRAMMES POUR AMSTRAD

Adaptation Alain Lavenir

EDITIONS DU P.S.I.
1986

SOMMAIRE

Avant-propos 8
Organisation de l'ouvrage 9

Niveau 1

Instructions élémentaires 12
Le nombre mystérieux 18
Cube 20
Chaud, froid 22
Les allumettes 24
Allumettes à plusieurs 26
Calendrier perpétuel 28
Bataille navale 30
Golf 32
Solfège 34
Le jeu du 21 36
Pair - Impair 38
Devine un chiffre 40
Le jeu de Witthof 42
Rebonds 44
Jackpot 46
Roulette 48
Couloirs 50
Echiquier 52
Jeu de la fève 54
Jeu de l'oie aléatoire 56

Niveau 2

Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et dans une chaîne - L'instruction DATA 60
Dé truqué 62
Un dé 64
Visions 66
421 68
L'électricien fou 70
La course au 2Ø 72
Musique 74
Poker 76
Le jeu de saute-mouton 78
Lettres 80
Machinamot 82

Espion 84
Jeu de loto 86
Tirage d'un dé truqué 88
Le jeu de Marienbad 90
Compose 92
Jour de la semaine 94
Tour de cartes 96
L'indésirable 98
Le jeu de la vie 100

Niveau 3

Manipulation des chaînes et graphiques haute résolution 104
Losanges 106
Plus grand, plus petit 108
Ordre 110
Le pendu 112
Belote 114
Un taquin de calcul 116
Pendu inversé 118
Nicomaque 120
Mastermind 122
Idem 124
Reines sur l'échiquier 126
Mots 128
Permutations 130
Bridge 132
Hanoï 134
Course de chevaux 136
Accrochez les wagons 138
Le loup et les agneaux 140

Niveau 4

Les techniques d'animation élémentaire 144
Tirage de dés 146
La bombe 148
Chapeaux 150
Dessin de dés 152
Le singe 154
Le jeu du 15 156
Clavier 158
Adresse 160
Anagrammes 162
Zombies 164
Ardoise magique 166
Décomaque 168

Evasion 170
Répétitions 172
Mathieu 174
A travers 176
Calculatrice 178
Chasse au sous-marin 181

Niveau 5

Les techniques particulières à l'AMSTRAD 185
Horloge 1 186
Horloge 2 188
Combat aérien 190
Affichage publicitaire 192
Le jeu du missile 194
Sous-marin 196
Tic Tac Toe 198
Smokey 200
Course de voitures 202
Danger spatial 204
Diago 206
Œufs 208
Taquin de lettres 210
Cibles 212
Gouttes 214
Obstacles 216
Captures dans l'espace 218
Anomalie 220
Plusieurs en un 222
Parcours du cavalier 224
Othello 226
Dames 229
Quatre à quatre 232
Echecs 235
Mur de briques 238
Le lézard d'Alice 240

Avant-propos

Vous trouverez dans ce livre plus de 100 programmes (essentiellement des jeux), rédigés en Basic AMSTRAD. La plupart de ces programmes seront d'ailleurs utilisables sans grande modification sur d'autres ordinateurs utilisant un Basic Microsoft.

Les programmes proposés sont très courts, et l'ordre dans lequel ils sont présentés n'est bien évidemment pas arbitraire : il s'agit d'initier le lecteur, à travers la réalisation et l'utilisation de programmes de jeux de bonne qualité, à la connaissance et à la maîtrise progressive de cet outil extraordinaire qu'est un ordinateur individuel.

La brièveté des programmes proposés permettra de les taper facilement, sans risque d'erreur. Vous pourrez bien sûr les utiliser comme base pour la réalisation de jeux plus performants.

Les programmes sont abondamment commentés pour vous aider à comprendre leur fonctionnement, et les pages de présentation vous permettront de retrouver facilement les caractéristiques propres des instructions Basic qu'ils utilisent.

Le niveau 5 donne une idée sur l'utilisation de certaines techniques un peu spécifiques. Ces techniques pourront être reprises avec profit pour améliorer les programmes précédents.

Vous trouverez, à la fin de chaque texte de présentation, quelques suggestions pour des améliorations possibles, ou des modifications, qui pourraient rendre le jeu proposé plus attrayant, plus performant ou plus complet.

Organisation de l'ouvrage

Les jeux sont classés par ordre de difficulté croissante, suivant 5 niveaux :

- **Le niveau 1** permet l'étude des instructions élémentaires du langage Basic : **LET** (optionnel ici), **PRINT, INPUT, IF...THEN, FOR...NEXT, GOTO, GOSUB, ON...GOTO, ON...GOSUB, WHILE...WEND.**

- **Au niveau 2,** on utilisera les instructions de confection et de manipulation des éléments d'un tableau, et de recherche d'un élément dans une chaîne : **DIM, R(I), MID$, READ, DATA.**

- **Au niveau 3,** on utilise la panoplie complète des instructions de traitement des chaînes de caractères : **MID$, RIGHT$, LEFT$.**

- **Le niveau 4** fait appel à une instruction très particulière qui facilite la réalisation de jeux interactifs : **INKEY$** et nous fait découvrir l'interaction du mode texte et du mode graphique avec **TAG**.

- Enfin, **le niveau 5** nous fait découvrir des instructions plus spécifiques : interruptions, lecture d'écran, générations de caractères.

Chaque jeu est présenté d'une façon très systématique, par une courte description, suivie de l'étude détaillée de chacune des lignes importantes du programme. Après quoi les indications nécessaires au bon fonctionnement du jeu sont données, ainsi que quelques éléments pouvant conduire à des extensions ou des améliorations du jeu proposé.

Le programme, écrit en Basic AMSTRAD, est ensuite reproduit, ainsi qu'un exemple d'exécution. Les listings ont été obtenus directement d'après les originaux de l'auteur, sur une imprimante matricielle, ce qui devrait garantir contre tout risque de disfonctionnement. Les minuscules ne sont pas utilisées. Par contre, rentrez les programmes en minuscules. Au listage les mots clés, et eux-seuls, s'impriment en majuscules. Les erreurs de frappe sont ainsi aisément détectées.

Avant de relancer tout programme, assurez-vous que vous avez réinitialisé la machine à l'aide de la suite d'instructions : **MODE 1: INK 0,1: INK 1,24: INK 2,20: INK 3,6.**

Attention, sur l'écran de l'Amstrad, les zéros apparaissent carrés et les "O" apparaissent ronds contrairement à ce qui ressort sur nos listings.

Niveau 1

Nom du programme	*Thèmes et particularités*
Le nombre mystérieux	Jeu de déduction, pour un joueur. Plusieurs niveaux.
Cube	Jeu de dessin : perspective sur un écran.
Chaud, froid	Logique et hasard, pour un joueur. Plusieurs niveaux.
Les allumettes	Jeu de réflexion, contre l'ordinateur.
Allumettes à plusieurs	Jeu de réflexion et de stratégie, pour plusieurs joueurs.
Calendrier perpétuel	Confection d'un calendrier, mois par mois.
Bataille navale	Un seul navire à trouver, mais quelques particularités.
Golf	Un parcours de 18 trous, pour un seul joueur.
Solfège	Apprenez le solfège à l'aide de votre ordinateur.
Le jeu du 21	Jeu de hasard, l'ordinateur tient la banque
Pair - Impair	Un jeu de casino en trois temps (un ou plusieurs joueurs).
Devine un chiffre	Un jeu très facile, pour un ou plusieurs joueurs.
Le jeu de Wytthof	Un jeu de réflexion assez simple, pour deux joueurs.
Rebonds	Un jeu d'adresse très visuel, pour un joueur.
Jackpot	Jeu de hasard, pour un joueur.
Roulette	Une simulation élégante du jeu très connu
Couloirs	Un jeu de logique, à découvrir.
Echiquier	Une base graphique pour vos propres créations.
Le jeu de la fève	Un jeu de réflexion, pour deux joueurs.
Jeu de l'oie aléatoire	Un très vieux jeu, revu et corrigé par l'ordinateur, deux joueurs.

Instructions élémentaires

Nous ne retiendrons ici que les instructions fondamentales, ou instructions de programmation ; beaucoup d'autres commandes, ou fonctions particulières, seront utilisées à l'occasion, par exemple :

– ***RND pour obtenir un nombre aléatoire entre*** **Ø et 1.**

– ***INT pour prendre la partie entière d'un résultat.***

– ***REM pour placer un commentaire.***

– ***END pour terminer un programme.***

– ***etc.***

Les instructions fondamentales de la programmation, utilisées dès le niveau 1, et présentes dans pratiquement tous les programmes sont :

LET (optionnelle) ; **PRINT ; INPUT ; GOTO ⌴ ; IF ⌴ THEN ⌴ ; FOR ... NEXT ; GOSUB ... RETURN.**

soit seulement sept instructions de base (et deux instructions dérivées) que nous allons maintenant examiner plus en détail.

L'INSTRUCTION LET OU INSTRUCTION D'AFFECTATION

C'est l'instruction par excellence des langages de programmation évoluée, puisque son rôle essentiel est de permettre, pour le programmeur, une gestion transparente de la mémoire. Cela signifie, en pratique, que pour ranger une quantité en mémoire, par exemple 12, il n'est pas nécessaire de savoir dans quelle case de la mémoire on va la déposer : il suffit de lui affecter un nom autorisé, et c'est par ce nom que la quantité sera par la suite retrouvée et manipulée.

– **Si la quantité est numérique,** le nom sera une simple lettre de l'alphabet, ou un ensemble de plusieurs chiffres ou lettres, commençant obligatoirement par une lettre.

– **Si la quantité est alphanumérique,** la même règle s'applique pour le nom, qui devra cependant être suivi du symbole $, et la variable devra être écrite entre guillemets.

Exemple :
LET A1 = 12 ; la variable numérique A1 prend la valeur 12
LET NOM $ = "RENOIR" ; la variable alphanumérique NOM$ prend la valeur RENOIR.

Sur AMSTRAD, comme dans la plupart des Basic, le LET est devenu optionnel, aucune ambiguïté n'étant possible, on écrira donc : A = 12 ; NOM$ = "TOTO" ; etc.

L'INSTRUCTION PRINT OU INSTRUCTION DE SORTIE (ECRAN)

C'est une instruction de dialogue, qui permet à la machine d'afficher sur l'écran un texte, le résultat d'un calcul, ou une variable. C'est le seul moyen que possède l'ordinateur individuel pour faire connaître ses résultats. Ainsi, PRINT A1 provoque l'affichage de 12 (si l'on a fait auparavant A1 = 12), PRINT A$ provoque l'affichage de BONJOUR (si l'on a fait A$ = "BONJOUR"). Utilisé après un effacement d'écran, le PRINT provoque l'affichage sur la 1re ligne, en haut, à gauche.

Un certain nombre de paramètres peuvent être utilisés pour obtenir sur l'écran un affichage convenable :

- **PRINT TAB(N)** provoque sur la ligne d'impression courante un affichage commençant à la énième colonne à partir de la gauche (**1** ⩽ N ⩽ 4Ø en mode 1).

- Un **; (point-virgule)** en fin de PRINT permet soit d'enchaîner une autre impression sans répéter l'instruction PRINT, soit d'indiquer que la prochaine impression sera, en l'absence d'indication contraire, effectuée exactement à la suite de l'impression précédente.

- La **,** (**virgule**) provoque une tabulation horizontale automatique prédéfinie modifiée par la **fonction ZONE**.

Lors de la frappe, **l'instruction PRINT** pourra être abrégée en tapant **?** (**point d'interrogation**). Lors d'une impression, tout nombre est précédé d'un signe ; mais le signe positif n'est pas affiché, un espace précédera donc automatiquement l'impression de tout résultat positif.

L'INSTRUCTION LOCATE

AMSTRAD dispose d'une instruction particulière, d'un emploi très souple, qui ***permet de positionner le curseur sur l'écran, en vue de la prochaine écriture.*** Il s'agit de l'instruction LOCATE X, Y. X correspond à une tabulation horizontale (entre 1 et 4Ø) et Y à une tabulation verticale (entre 1 et 25) en mode texte.

Ainsi, pour écrire le mot BONJOUR à la 5e colonne de la 8e ligne, on écrira :

LOCATE 5,8 : PRINT "BONJOUR"

L'INSTRUCTION INPUT OU INSTRUCTION D'ENTREE (CLAVIER)

Cette instruction, qui provoque un arrêt et une attente de l'appareil, ***permettra d'entrer un nombre, ou un texte, qui sera affecté à la variable suivant l'ordre INPUT.*** Ceci est fait en mode conversationnel, c'est-à-dire au fur et à mesure des besoins ; en effet, lorsque l'AMSTRAD rencontre une instruction INPUT, il s'arrête et vous rend la main ; on peut alors lui entrer une information de nature convenable (nombre ou texte), et lui signaler que cette entrée est terminée à l'aide de la touche **ENTER,** ce qui lui rendra la main.

Si l'information entrée n'est pas de nature convenable (vous tapez, par exemple, un texte sur un INPUT X, qui attend un nombre), le message **REDO FROM START** est affiché, et l'appareil se remet en attente.

Il est possible, comme pour le PRINT, d'afficher des messages avant l'entrée de la variable, et d'utiliser l'instruction LOCATE.

On pourra, par exemple, écrire :

INPUT "VOTRE NOM, S'IL VOUS PLAIT" ; A$

LES FENETRES D'ECRAN

L'écran peut être divisé en huit fenêtres, dont chacune a sa couleur d'encre et sa position de curseur. Par défaut, les 8 fenêtres sont rigoureusement superposées. L'ordre WINDOW permet de les limiter à une partie de l'écran. Ainsi, WINDOW # 1, 2Ø, 25, 12, 14 ouvre une fenêtre d'écran entre les colonnes 2Ø et 25 et les lignes 12 et 14.

Les instructions PRINT, INPUT et LOCATE, pour être exécutées dans une fenêtre particulière, doivent utiliser son numéro.

Ainsi PRINT # 1, "TOTO" imprimera sur la première fenêtre, aux coordonnées (éventuellement) définies par un ordre du type LOCATE # 1, X, Y. En présence d'une fenêtre, X et Y sont comptés par rapport au coin supérieur gauche de celle-ci, indépendamment de la position sur l'écran habituel 4Ø × 25 lignes.

L'INSTRUCTION GOTO OU INSTRUCTION DE RENVOI INCONDITIONNEL

Une des particularités du langage Basic, qui le rend facilement accessible, est que ***chaque ligne de programme doit être précédée d'un numéro,*** ces lignes étant ensuite exécutées par l'ordinateur individuel en séquences, dans le sens des numéros croissants. Pour diverses raisons, il pourra être nécessaire au cours du déroulement d'un programme, de ne pas suivre cet ordre séquentiel très rigide ; on utilisera alors ***l'instruction GOTO, qui permet de détourner le fonctionnement du programme de son cours nor-***

mal quitte d'ailleurs à le reprendre ensuite par un autre GOTO. Le renvoi doit, par ailleurs, impérativement se faire sur un numéro de ligne existant dans le programme (GOTO 2Ø, par exemple).

Pour donner un peu de souplesse à cette instruction de renvoi, il existe une instruction très puissante, qui permet de renvoyer à n'importe quel numéro de ligne, selon la valeur d'une variable numérique : **c'est l'instruction ON⊔ GOTO⊔** ; on écrira par exemple :

ON X GOTO 1ØØ, 2ØØ, 3ØØ et le renvoi sera fait à la ligne 1ØØ si X vaut 1, à la ligne 2ØØ si X vaut 2, et 3ØØ si X vaut 3.

L'INSTRUCTION IF⊔ THEN⊔ OU INSTRUCTION DE TEST

On peut l'utiliser de deux façons : d'abord, en rupture de séquence conditionnelle, sous la forme :

IF condition = vraie THEN numéro de ligne.

C'est l'usage le plus courant de ce test, qui permet de renvoyer à une autre partie de programme, si par exemple X = Ø, ou si A = B, etc. Les **opérateurs logiques NOT, OR, XOR** ou **AND** peuvent d'ailleurs être utilisés dans l'écriture de la condition.

On pourra aussi utiliser le test sous la forme :

IF condition THEN instruction ;

l'instruction présentée à la suite de THEN peut être une affectation A = 12, une impression PRINT A, une entrée INPUT B $, ou même un autre test IF ⊔ THEN ⊔.

On peut par exemple écrire :

IF A > 3 THEN IF B < 7 THEN PRINT A + B

L'INSTRUCTION FOR...NEXT OU INSTRUCTION DE BOUCLES

On est très souvent conduit, en programmation, à vouloir exécuter un nombre entier de fois une certaine suite d'instructions (par exemple, faire avancer un mobile de la case 6 à la case 2Ø, etc.). Plutôt que d'utiliser un test et un renvoi, il sera préférable de faire appel à une ***boucle,*** la variable de boucle prenant ses valeurs dans les entiers ou les décimaux, et le ***pas*** de cette variable (c'est-à-dire la quantité dont elle augmente ou diminue à chaque passage) pouvant aussi prendre ses valeurs dans les entiers ou les décimaux.
Ainsi, si l'on écrit :

```
FOR I = }Ø TO 2Ø STEP 2
NEXT I  }Instructions
```

Les instructions contenues dans la boucle seront exécutées pour I = Ø, pour I = 2, ... et ceci jusqu'à I = 2Ø, soit 11 fois. La variable I peut être utilisée pour faire un calcul dans la boucle, mais ne devra pas être modifiée dans la boucle. On peut, bien sûr, sortir de la boucle, avant I = 2Ø, par un test portant une quantité quelconque.

Si ***le pas de la boucle*** (STEP) n'est pas précisé, il sera automatiquement pris égal à 1.

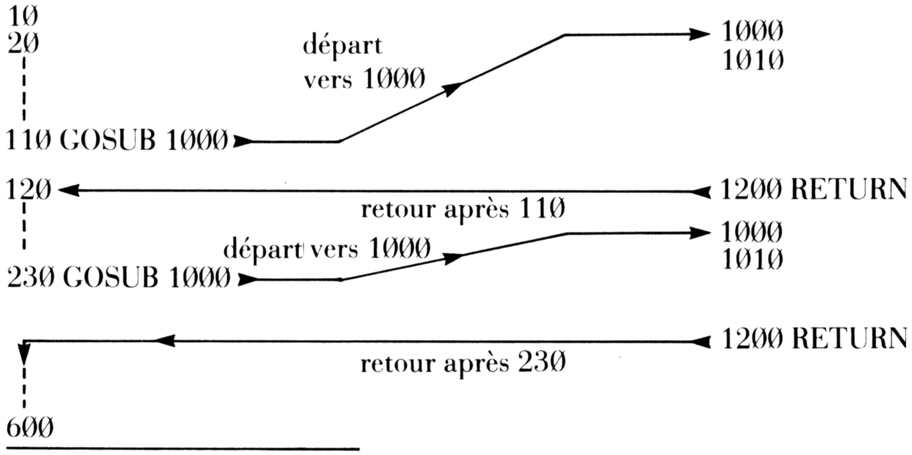
Il est possible d'emboîter plusieurs boucles de noms différents, et d'omettre l'indication de la variable dans le **NEXT.**

L'INSTRUCTION GOSUB OU INSTRUCTION D'APPEL DE SOUS-PROGRAMME

Lorsqu'une même suite d'instructions doit être exécutée plusieurs fois, et appelée à partir de plusieurs endroits du programme, on évitera de la réécrire plusieurs fois. On la rédige une fois pour toutes, et on la place en dehors du programme proprement dit (en lui donnant, par exemple, un numéro de ligne supérieur au numéro de la dernière ligne du programme). Indépendamment des difficultés que l'on pourra rencontrer au niveau du nom des variables, lorsque l'on sera désireux d'utiliser cette séquence, il suffira de l'appeler par **l'instruction GOSUB numéro de ligne.** Le programme ira dès lors "faire un détour" dans la partie du programme appelé, mais il ne pourra revenir et poursuivre son déroulement normal que si il rencontre dans le programme appelé une instruction **RETURN,** ou instruction de retour.

Voici le schéma théorique de fonctionnement :

Programme principal



Fin du programme principal

1ØØØ

12ØØ RETURN

Sous-programme 1

Le sous-programme n'est écrit qu'une fois, mais appelé aussi souvent que l'on veut.

Un sous-programme pourra lui-même éventuellement appeler un autre sous-programme, et ainsi de suite.

On peut aussi appeler un sous-programme de façon conditionnelle, en utilisant l'instruction ON␣ GOSUB ␣.

Par exemple ON X GOSUB 1ØØØ, 2ØØØ, 3ØØØ permettra d'exécuter le sous-programme placé en 1ØØØ si X = 1, en 2ØØØ si X = 2, en 3ØØØ si X = 3.

L'INSTRUCTION WHILE/WEND

Lorsqu'une suite d'instructions doit être exécutée tant qu'une certaine condition est vérifiée, il suffit de la mettre au milieu d'une boucle WHILE/WEND. **Attention, avant de commencer toute boucle WHILE/WEND, il est prudent d'initialiser la (ou les) variable(s) utilisée(s) pour le test de condition,**

EXEMPLE :

```
1Ø WHILE TE < 2ØØ
2Ø TE = TE + 1 : PRINT TE
3Ø WEND
```

La variable TE est augmentée d'une unité et imprimée tant qu'elle reste inférieure à 2ØØ.

Il est possible d'emboîter plusieurs boucles WHILE/WEND entre elles (programme HORLOGE 2) ce qui permet une grande élégance dans l'écriture des programmes.

Le nombre mystérieux

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction pour un joueur.
Difficulté du jeu : facile.

LE JEU :
On demande à l'ordinateur de fabriquer un nombre de n chiffres (avec n compris entre 1 et 7).
On devra ensuite essayer de le découvrir, avec le moins d'essais possible, en recevant les seuls renseignements TROP GRAND ou TROP PETIT.

LE PROGRAMME :
La ligne 6Ø permet de fabriquer le nombre à deviner.
Les entrées sont faites par les lignes 8Ø et 9Ø.
La ligne 1ØØ teste si le résultat est exact.
La ligne 11Ø signale un essai trop petit, et la ligne 12Ø un essai trop grand.
Le compteur d'essais est en ligne 13Ø.
Les lignes 16Ø à 18Ø s'occupent de l'affichage du résultat, et de la gestion de la partie suivante.

POUR JOUER :
On lancera le programme par RUN.
On obtient un premier affichage NOMBRE DE CHIFFRES A DEVINER auquel on répond par un entier entre 1 et 7, suivi de ENTER.
On obtient alors l'affichage ESSAI N° 1, suivi de la question VOTRE NOMBRE.
On entre un nombre, et l'on presse ENTER.
On obtient alors une réponse TROP PETIT, ou TROP GRAND, ou encore, si l'on a trouvé juste, un affichage de VOUS AVEZ TROUVE... EN... ESSAIS. Si le résultat n'est pas exact, on revient à ESSAI N° 2, et VOTRE NOMBRE.
En fin de partie, on obtient la question AUTRE PARTIE (O/N) ; on répond O ou **o** pour faire une autre partie (suivi de ENTER).
On peut arrêter à tout moment en tapant deux fois de suite sur la touche ESC.

EXTENSIONS POSSIBLES :
On pourra imaginer de faire deviner au joueur autre chose qu'un chiffre (par exemple, un mot, en utilisant l'ordre alphabétique, ou encore autre chose...).

```
10 REM LE NOMBRE MYSTERIEUX
20 CLS
30 INPUT "NOMBRE DE CHIFFRES A DEVINER:
 ";N
40 IF N>7 THEN RUN
50 IF N<=0 THEN RUN
60 X=INT(RND(1)*10↑N+1)
70 J=1
80 PRINT:PRINT"ESSAI NUMERO ";J

90 INPUT"VOTRE NOMBRE :";M
100 IF M=X THEN 160
110 IF M<X THEN PRINT "TROP PETIT"
120 IF M>X THEN PRINT "TROP GRAND"
130 J=J+1
140 PRINT
150 GOTO 80
160 PRINT:PRINT "VOUS AVEZ TROUVE EN ";J
;" ESSAIS"
170 INPUT "UNE AUTRE PARTIE (O/N)";R$
180 IF R$="O" OR R$="o" THEN RUN
```

Cube

REFERENCES :
Difficulté de programmation : aucune.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
Le programme proposé se contente de réaliser à l'aide d'instructions PRINT une vue d'un cube en perspective cavalière.
Il pourrait être par exemple incorporé à un jeu basé sur les capacités de reconnaissance des formes géométriques élémentaires.

LE PROGRAMME :
Le dessin du cube est réalisé par l'impression de points, en utilisant deux boucles : la boucle I (30-120), et la boucle I (130-180).
Le nom de l'objet représenté est affiché en ligne 200 (ici : CUBE).

POUR JOUER :
Il suffit de lancer l'exécution par RUN, et de regarder. On fera ESC deux fois pour reprendre la main.

EXTENSIONS POSSIBLES :
La haute résolution permettra d'avoir un tracé plus précis.
On pourra imaginer un jeu utilisant des parallélépipèdes rectangles de différentes tailles, avec affichage par exemple des 3 côtés, et il faudrait trouver leur volume, etc.

```
10 REM CUBE
20 CLS
30 FOR I=1 TO 11
40 LOCATE 7+I,1:PRINT "."
50 LOCATE I,8:PRINT "."
60 LOCATE 8+I,12:PRINT "."
70 LOCATE 1+I,19:PRINT "."
80 LOCATE 1,8+I:PRINT "."
90 LOCATE 12,7+I:PRINT "."
100 LOCATE 8,I+1:PRINT "."
110 LOCATE 19,I:PRINT "."
120 NEXT I
130 FOR I=1 TO 6
140 LOCATE 1+I,8-I:PRINT "."
150 LOCATE 12+I,8-I:PRINT "."
160 LOCATE 1+I,19-I:PRINT "."
170 LOCATE 12+I,19-I:PRINT "."
180 NEXT I
190 LOCATE 26,19:PRINT CHR$(150);STRING$
(4,154);CHR$(156)
200 LOCATE 26,20:PRINT CHR$(149);"CUBE";
CHR$(149)
210 LOCATE 26,21:PRINT CHR$(147);STRING$
(4,154);CHR$(153)
220 GOTO 220
```

Chaud, froid

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Un des plus vieux jeux du monde, toujours aussi pratiqué, mais cette fois, c'est l'ordinateur qui vous sert de partenaire : il va en effet cacher un objet à un certain emplacement d'une grille dont vous aurez vous-même choisi les dimensions. Pour le trouver, il ne vous donnera que quelques indications sous la forme BRULANT, ou COMPLETEMENT GELE, etc.
Vous devez réussir à découvrir l'objet caché, avec le moins d'essais possible (votre score sera affiché en fin de partie).

LE PROGRAMME :
Les lignes 4Ø et 5Ø définissent la position de l'objet.
La ligne 1ØØ calcule la distance par rapport aux deux coordonnées que vous avez entrées aux lignes 7Ø et 8Ø.
La ligne 11Ø regarde si vous avez gagné.
Les lignes 13Ø-14Ø normalisent la distance (qui doit avoir un sens, quelles que soient les dimensions de la grille initiale).
La ligne 15Ø effectue le départ, s'il y a lieu, vers l'un des sous-programmes d'affichage.

POUR JOUER :
On lance le programme par l'ordre RUN, puis on introduit les dimensions de la grille (carrée) et l'on frappe ENTER.
Dès l'affichage suivant (DONNEZ X), on entre la coordonnée X supposée, suivie de ENTER.
On obtient ensuite l'affichage DONNEZ Y, et l'on entre Y suivi de ENTER.
On obtient en réponse un texte, qui est variable selon que l'on sera plus ou moins loin de l'objet cherché (cela va de COMPLETEMENT GELE à BRULANT). Lorsque l'on a trouvé, on obtient l'affichage du nombre d'essais effectués.
Pour sortir du programme avant la fin de la partie, il suffit de taper ESC deux fois de suite.

EXTENSIONS POSSIBLES :
Si l'on dispose de la couleur, il sera intéressant de trouver un moyen plus visuel pour symboliser la distance, par exemple des pavés allant du rouge au bleu, aux emplacements essayés.
On peut aussi travailler dans un espace à trois dimensions, cacher plusieurs objets, etc.

```
10 REM CHAUD FROID
20 CLS
30 INPUT "DIMENSION DE LA GRILLE: ";D
40 X=INT(RND*D+1)
50 Y=INT(RND*D+1)
60 CLS
70 INPUT "DONNEZ X: ";A
80 INPUT "DONNEZ Y: ";B
90 S=S+1
100 T=ABS(X-A)+ABS(Y-B)
110 IF T=0 THEN 280
120 PRINT:PRINT"VOUS ETES ";
130 N=INT(SQR(D)/T)+1
140 IF N>5 THEN N=5
150 IF T<=SQR(D) THEN ON N GOTO 180,200,
220,240,260
160 PRINT"COMPLETEMENT GELE."
170 GOTO 70
180 PRINT"A PEINE TIEDE."
190 GOTO 70
200 PRINT"TIEDE."
210 GOTO 70
220 PRINT"CHAUD."
230 GOTO 70
240 PRINT"TRES CHAUD."
250 GOTO 70
260 PRINT"BRULANT"
270 GOTO 70
280 CLS:BORDER 3,4
290 LOCATE 10,12:PRINT "TROUVE EN ";S;"
ESSAIS."
300 FOR T=1 TO 5000:NEXT T
310 BORDER 1,1
```

Les allumettes

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : contre AMSTRAD.
Difficulté du jeu : moyenne.

LE JEU :
Le joueur devra en début de partie choisir une certaine quantité d'allumettes, ainsi que la prise maximum autorisée. Il décide ensuite s'il commence, ou s'il laisse commencer AMSTRAD. Chacun prend à son tour autant d'allumettes qu'il le désire, dans la limite du nombre maximum autorisé.
Celui qui prend la dernière allumette est considéré comme perdant.

LE PROGRAMME :
Le jeu du joueur commence en ligne 7Ø.
Celui d'AMSTRAD commence en ligne 11Ø.
La ligne 8Ø contrôle la légalité du coup du joueur.
La ligne 9Ø gère le changement de joueur.
Le coup d'AMSTRAD est calculé dans une boucle non formelle, de la ligne 12Ø à la ligne 15Ø.
Le résultat est affiché par la ligne 1Ø2Ø.
Si la partie est terminée, la ligne 1Ø3Ø le détecte et envoie en ligne 2ØØØ, pour le message de fin.

POUR JOUER :
On commence l'exécution par RUN, et, à l'affichage du message NOMBRE TOTAL, on entre le nombre d'allumettes, suivi de ENTER. Puis on entre la prise maximale autorisée, suivie de ENTER. Si le joueur veut commencer, il frappe ensuite Ø ; sinon il frappe 1, puis ENTER.
Après chaque coup, le nombre d'allumettes restantes est affiché, et AMSTRAD affiche VOUS RETIREZ lorsque c'est au joueur de prendre.
Celui qui prend la dernière allumette a perdu.

EXTENSIONS POSSIBLES :
On pourra bien sûr envisager un affichage graphique couleur plus spectaculaire, imaginer de laisser le hasard choisir le nombre d'allumettes et la prise maximum, etc.

```
10 REM LES ALLUMETTES
20 CLS
30 INPUT "NOMBRE TOTAL  :";N
40 INPUT "PRISE MAXIMUM :";I
50 INPUT "QUI COMMENCE (VOUS:0 / MOI:1)
:";D
60 IF D=1 THEN 110
70 INPUT "VOUS RETIREZ :";P
80 IF P<>INT(P) OR P<1 OR P>N OR P>I THE
N 70
90 D=1
100 GOTO 1000
110 S=I+D
120 T=(N-S)/(I+D)
130 IF T=INT(T) THEN 200
140 S=S-D
150 GOTO 120
200 P=ABS(S-D)
210 IF P=0 THEN P=D
220 D=0
1000 PRINT P
1010 N=N-P
1020 PRINT "IL RESTE ";N
1030 IF N=0 THEN 2000
1040 GOTO 60
2000 CLS
2010 IF D=1 THEN PRINT "J'AI GAGNE":SOUN
D 1,478:SOUND 1,426:SOUND 1,379
2020 IF D<>1 THEN PRINT "BRAVO":SOUND 1,
256,4:SOUND 1,256,4,0:SOUND 1,256,4
```

Allumettes à plusieurs

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : logique (plusieurs joueurs).
Difficulté du jeu : assez difficile.

LE JEU :
Dans ce jeu pour plusieurs joueurs, le meneur de jeu devra, en début de partie, choisir un nombre d'allumettes, et indiquer à votre AMSTRAD le nombre de joueurs.
A chaque coup, le joueur qui a le trait est autorisé à retirer du tas d'allumettes restantes jusqu'au double de ce que vient de retirer le précédent joueur ; celui qui retire la dernière allumette a gagné (ou perdu, si l'on modifie légèrement les règles).

LE PROGRAMME :
La ligne 20 permet de travailler sur des entiers en provoquant l'arrondi des valeurs.
Les lignes 40 à 60 permettent les différentes entrées.
Les lignes 100 à 120 réalisent les impressions.
La ligne 140 contrôle le coup de chaque joueur (par rapport au coup précédent).
La ligne 200 gère les tours (remise à 1 du numéro du joueur).
La ligne 180 réalise le test de fin, et la ligne 230 affiche le numéro du joueur gagnant (avec la règle prévue initialement).

POUR JOUER :
Après avoir lancé le programme par RUN, on entre le nombre initial d'allumettes, suivi de ENTER, puis le nombre de joueurs (et ENTER).
Le premier joueur ne pourra enlever qu'une allumette, le deuxième pourra en enlever 1 ou 2 ; le suivant 1, 2 ou 1, 2, 3 ou 4 selon le coup précédent, et ainsi de suite ; le retrait maximum autorisé est à chaque fois affiché par AMSTRAD.
Lorsque la dernière allumette a été prise, on obtient l'affichage LE JOUEUR... GAGNE, et la partie est terminée.

EXTENSIONS POSSIBLES :
On peut convenir que celui qui prend la dernière allumette est éliminé du jeu, et recommencer la partie entre les joueurs restants qui seront ainsi éliminés les uns après les autres.
On peut aussi imaginer une représentation graphique des tas d'allumettes, tirer au hasard le nombre à enlever en début de partie, etc.

```
10 REM allumettes a plusieurs
20 DEFINT F,J,K,N
30 CLS
40 INPUT "NOMBRE D'ALLUMETTES:";N
50 PRINT
60 INPUT "NOMBRE DE JOUEURS   :";J
70 F=1:K=1:INK 1,0
80 CLS
90 INK 0,K
100 PRINT TAB(12)"TOUR DU JOUEUR";K:PRIN
T
110 PRINT "TOTAL RESTANT      :";N
120 PRINT "MAXIMUM A ENLEVER :";F
130 INPUT"VOTRE COUP          :  ",C
140 IF C<=0 OR C>F THEN 130
150 N=N-C
160 IF N<0 THEN N=N+C:GOTO 130
170 F=C+C
180 IF N=0 THEN 220
190 K=K+1
200 IF K>J THEN K=1
210 GOTO 80
220 CLS
230 LOCATE 12,12:PRINT "LE JOUEUR";K;"GA
GNE."
```

Calendrier perpétuel

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : gadget publicitaire.
Difficulté du jeu : aucune.

LE JEU :
Ce programme vous permettra de confectionner tous les ans votre calendrier ; vous pourrez ainsi vous en servir pour envoyer à vos amis un calendrier du mois de leur naissance, ou pour leur souhaiter une bonne fête, etc.

LE PROGRAMME :
Les lignes 3Ø et 5Ø vous permettront d'entrer l'année et le mois.
Les lignes 6Ø, 7Ø et 8Ø calculent le numéro du premier jour du mois.
La ligne 9Ø calcule le nombre de jours du mois (il n'est pas tenu compte ici des 29 février, pour simplifier).
La boucle A (16Ø-19Ø) réalise l'impression du calendrier proprement dit.
La ligne 14Ø, à l'aide de l'ordre **ZONE**, permet d'espacer de 5 blancs les éléments de liste séparés par une virgule.
La ligne 2ØØ évite l'apparition du mot **READY**.

POUR JOUER :
Après avoir lancé le programme par l'instruction RUN, on devra introduire l'année, sous la forme, par exemple 1985, suivie de ENTER.
On obtient alors la question MOIS, à laquelle il convient de répondre par le numéro du mois, suivant l'ordre usuel (1 pour JANVIER, 2 pour FEVRIER, etc.).
On fait ensuite ENTER, et le calendrier est alors affiché sur l'écran.

EXTENSIONS POSSIBLES :
On pourra sans difficulté prévoir l'affichage des 29 février, prévoir une option permettant l'affichage pour une année complète, prévoir l'introduction du nom des mois en clair, etc.

```
10 REM CALENDRIER PERPETUEL
20 CLS
30 INPUT "POUR QUELLE ANNEE :";A
40 PRINT
50 INPUT "POUR QUEL MOIS     :";M
60 J=INT(365.25*(A+(M<3)))+1
70 J=J+INT(30.6*(M+1-(M<3)*12))-INT((INT
((A+(M<3))/100)-7)*0.75)
80 J=J-7*INT(J/7)
90 M=31+(M=4 OR M=6 OR M=9 OR M=11)+(M=2
)*3
100 PRINT:PRINT:PRINT
110 PEN 2
120 PRINT "DIM  LUN  MAR  MER  JEU  VEN
 SAM"
130 PRINT
140 ZONE 5:PEN 3
150 LOCATE 5*J+1,10
160 FOR A=1 TO M
170 PRINT A,
180 IF J+A-1=6 THEN PRINT:PRINT:J=J-7
190 NEXT A
200 GOTO 200
```

Bataille navale

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

LE JEU :
Une bataille navale a priori assez simple, avec un seul navire caché au hasard dans une grille de 10 cases sur 10 cases.
A chaque coup, on devra tirer en donnant les coordonnées H (horizontale) et V (verticale) de la case visée, et AMSTRAD répond en affichant la distance séparant le navire invisible de la case visée. Mais, si l'on tire trop près, le navire risque de se recacher au hasard.

LE PROGRAMME :
La boucle I (50-80) dessine le terrain de jeu.
Les lignes 100-110 définissent l'emplacement du navire.
Les coordonnées du tir sont entrées aux lignes 130 et 140, et la ligne 150 affiche le coup sur l'écran. La distance est calculée en ligne 160, et la ligne 170 regarde si l'on a ou non fait mouche.
La ligne 190 intervient pour déplacer le navire si les coups sont rapprochés.
La ligne 200 permet l'affichage de la distance.

POUR JOUER :
Le terrain de jeu est affiché après avoir frappé RUN. A l'affichage de LIGNE, on répond en donnant le numéro de la ligne horizontale sur laquelle on désire tirer. On tape ensuite ENTER et, à la question COLONNE, on répond en donnant le numéro de la colonne verticale suivi de ENTER.
Si on a tiré trop près, on obtient l'affichage PFFFF... qui signifie que le navire ennemi a changé d'emplacement.
Si on a tiré sur la bonne case, on obtient l'affichage du nombre total de tirs effectués pour gagner.

EXTENSIONS POSSIBLES :
Elles sont innombrables, puisqu'aussi bien la bataille navale est l'un des jeux les plus répandus qui soit. On peut introduire deux joueurs, jouer contre AMSTRAD, mettre plusieurs navires, des sous-marins, etc.

```
10 REM BATAILLE NAVALE
20 RANDOMIZE TIME
30 CLS
40 PRINT "   0  1  2  3  4  5  6  7  8
9":PRINT
50 FOR I=0 TO 9
60 PRINT  I;".  .  .  .  .  .  .  .  .
."
70 PRINT
80 NEXT I
90 S=1
100 X=INT (RND*10)
110 Y=INT (RND*10)
120 LOCATE 1,23:PRINT "
                      "
130 LOCATE 1,23:INPUT"LIGNE :";H
140 LOCATE 24,23:INPUT"COLONNE :";V
150 LOCATE 3*V+3,2*H+3:PEN 3:PRINT S:PEN
 1
160 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V)))
170 IF D=0 THEN 210
180 S=S+1
190 IF D=1 THEN LOCATE 1,24:PRINT "PFFF.
........          ": GOTO 100
200 LOCATE 1,24:PRINT "DISTANCE       :"
;D:GOTO 120
210 CLS
220 PRINT "TROUVE EN";S;"COUPS."
```

Golf

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez facile.

LE JEU :
Il s'agit d'essayer de réaliser, avec le moins de coups possible, un parcours de golf de 18 trous. Pour cela, il vous faut frapper la balle avec une force plus ou moins grande, jusqu'à obtenir sa chute dans le trou.
Mais attention, il peut y avoir un certain aléa sur chaque coup, et vous serez amené souvent à revenir en arrière (il faudra alors faire précéder l'indication de la force par un signe −).

LE PROGRAMME :
La ligne 7Ø permet de définir la position du trou, et la ligne 8Ø dessine le terrain de jeu.
Le déplacement à réaliser est calculé en ligne 12Ø.
La boucle I (15Ø-21Ø) réalise le déplacement de la balle.
La ligne 22Ø regarde si le coup est, ou non, gagnant ; s'il l'est, on recommence (ligne 27Ø) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (4Ø-27Ø) ; fin de boucle en 29Ø, avec affichage de fin).

POUR JOUER :
On obtient l'affichage du terrain en frappant RUN ainsi que le dessin de la balle sur sa position de départ, et le texte : Trou 1. Force ; on doit alors répondre en donnant la force désirée (un nombre entre 1 et 3ØØ...).
Dès que l'on aura tapé ENTER, la balle recommence à se déplacer ; elle peut alors soit s'arrêter avant le trou (auquel cas on recommence), soit tomber dedans (on passe alors au trou suivant), soit le dépasser : on devra alors frapper avec une force négative.
Le jeu se termine avec l'affichage 18 TROUS EN ... ESSAIS.

EXTENSIONS POSSIBLES :
Il sera possible d'étendre le jeu pour faire participer plusieurs joueurs, d'améliorer la qualité graphique, d'utiliser une représentation spatiale du terrain (se référer à certains jeux d'arcade), d'avoir un terrain plus accidenté, des obstacles, de changer de club...

```
10 REM GOLF
20 RANDOMIZE(TIME)
30 INK 0,9:INK 1,18:INK 2,0:INK 3,26
40 FOR S=1 TO 18
50 CLS
60 X=1
70 T=INT(29*RND+9)
80 PEN 1:LOCATE 1,12:PRINT STRING$(40,20
8)
90 PEN 3:LOCATE T,12:PRINT" ":LOCATE 1,1
1:PRINT "O";
100 PEN 2:LOCATE 1,17:PRINT "TROU";S;" F
ORCE ?"
110 LOCATE 16,17:PRINT "             "
120 PEN 2:LOCATE 16,17:INPUT "",R
130 F=INT(R*0.16-(ABS(R)*(SQR(RND)+0.1)↑
2)/1000)
140 IF F =0 THEN 110
150 FOR I=1 TO MIN (ABS(F),40)
160 X=X+SGN(F)
170 PEN 3
180 IF X<=40 AND X>=1 THEN LOCATE X-SGN(
F),11:PRINT " ":LOCATE X,11:PRINT "O"
190 IF X>=40 THEN X=40
200 IF X<=1 THEN X=1
210 NEXT I
220 IF X=T THEN 250
230 E=E+1
240 GOTO 110
250 LOCATE X,11:PRINT " ":LOCATE X,12:PR
INT "O"
260 FOR P=1 TO 1000:NEXT P
270 NEXT S
280 CLS
290 LOCATE 10,12:PRINT"18 TROUS EN";E;"E
SSAIS"
```

Solfège

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

LE JEU :
Ce programme vous propose une initiation élémentaire au solfège, en vous permettant d'apprendre à reconnaître facilement les différentes notes, qui apparaissent les unes après les autres sur une portée : après l'affichage de la note, vous devez entrer son nom en clair et votre score sera affiché en permanence.
Le programme est initialement prévu pour la clé de sol, mais il sera très facilement adaptable à toute autre clé.

LE PROGRAMME :
La boucle I (5Ø-8Ø) dessine une portée.
La note est choisie en ligne 1ØØ et affichée en lignes 13Ø à 16Ø avec encre transparente (ligne 12Ø).
La ligne 19Ø affiche le score.
La ligne 2ØØ affiche le nom de la clé (ici clé de sol) et vous interroge.
La ligne 22Ø s'occupe de la fin de la partie.
Les lignes 23Ø et 24Ø effectuent le décodage de la note entrée.
La ligne 26Ø contrôle le résultat.

POUR JOUER :
Après avoir lancé le programme par RUN, la portée sera affichée, ainsi que la première note.
Vous répondrez à la question NOTE en donnant le nom en clair de la note (do, ré, mi, fa, sol, la ou si), suivi de ENTER.
Le score vous indiquera si c'est exact ou non, et une nouvelle note sera affichée.
Pour arrêter, il suffira de répondre Ø suivi de ENTER, à la question NOTE.

EXTENSIONS POSSIBLES :
On pourra tout d'abord facilement modifier le programme pour lui permettre de vous faire travailler les différentes clés. On pourra aussi améliorer le graphisme, introduire la couleur, le son (lorsque l'on a répondu juste, on entendra la note, par exemple). On pourra aussi réaliser suivant le même principe un jeu vous permettant de reconnaître des accords, etc.

```
10 REM SOLFEGE
20 RANDOMIZE(TIME)
30 S=0:P=0
40 CLS
50 FOR I=1 TO 5
60 PEN 2
70 LOCATE 1,2*I+3:PRINT STRING$(40,210)
80 NEXT I
90 PEN 3
100 N=INT(RND*12)+1
110 X=N+3
120 PRINT CHR$(22)+CHR$(1)
130 LOCATE 21,X-2:PRINT CHR$(211):LOCATE
 21,X-2:PRINT CHR$(92)
140 LOCATE 21,X-1:PRINT CHR$(211)
150 LOCATE 19,X:PRINT CHR$(214);CHR$(215
);CHR$(211)
160 LOCATE 19,X+1:PRINT CHR$(213);CHR$(2
12)
170 PRINT CHR$(22)+CHR$(0)
180 PEN 1
190 LOCATE 10,21:PRINT"SCORE";S;"SUR";P;
"ESSAIS":PRINT
200 INPUT "CLE DE SOL - QUELLE EST LA NO
TE ";R$
210 R$=UPPER$(R$)
220 IF R$="O" THEN END
230 Y=(1 AND R$="SOL")+(2 AND R$="FA")+(
3 AND R$="MI")+(4 AND R$="RE")
240 Y=Y+(5 AND R$="DO")+(6 AND R$="SI")+
(7 AND R$="LA")
250 P=P+1
260 IF Y=N OR (Y=N-7 AND N>9)THEN S=S+1
270 GOTO 40
```

Le jeu du 21

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu contre AMSTRAD.
Difficulté du jeu : facile.

LE JEU :
On devra entrer une mise au début du jeu, et on pourra lancer les dés en tapant 1 ; à chaque lancer, on obtiendra un nouveau total, qui augmentera peu à peu, et se rapprochera de 21. On devra essayer de se rapprocher le plus possible de 21, sans dépasser ce nombre. Lorsque l'on décidera de s'arrêter, il suffira de taper Ø pour laisser jouer AMSTRAD qui essaiera alors d'avoir un meilleur score.
A la fin de chaque partie, le gain sera affiché ainsi que le total des gains (ou des pertes).

LE PROGRAMME :
Les lignes 6Ø et 7Ø réalisent le contrôle de celui qui doit jouer (joueur ou AMSTRAD).
Le sous-programme (41Ø-42Ø) lance les dés.
Les lignes 9Ø à 11Ø calculent le total fait par le joueur et l'affichent.
Le total d'AMSTRAD est calculé en ligne 18Ø et affiché par la ligne 2ØØ.
La ligne 23Ø contrôle la fin du jeu pour AMSTRAD.
Les lignes 26Ø-28Ø affichent les résultats.
Le gain (ou la perte) sera affiché par la ligne 33Ø, et le total par la ligne 36Ø.
La ligne 39Ø contrôle la fin de partie.

POUR JOUER :
Taper RUN pour lancer le jeu.
A la question VOTRE MISE, entrer la mise initiale, suivie de ENTER, et taper 1 pour lancer le dé. Le premier tirage est affiché sous la forme VOUS AVEZ... et on devra refaire 1 suivi de ENTER, tant que l'on désire recommencer à tirer.
Dès que l'on a terminé, on tape Ø et ENTER pour laisser jouer AMSTRAD (qui tient ici le rôle de la banque).
Le gain (ou la perte, s'il y a le signe −) sera affiché en fin de jeu, ainsi que le total des gains (ou pertes) réalisé depuis le début. Si l'on désire arrêter de jouer, on devra taper N en réponse à la question ON CONTINUE (O/N).

EXTENSIONS POSSIBLES :
Ce jeu sera évidemment plus intéressant s'il est pratiqué à plusieurs, et il sera aisé de modifier le programme en conséquence. On pourra aussi dessiner le dé, ou mieux, utiliser des cartes, ce qui permettra alors de faire un jeu de Black Jack.

```
10 REM JEU DU 21
20 RANDOMIZE(TIME)
30 CLS
40 T=0
50 INPUT "VOTRE MISE :";M
60 LOCATE 3,25:INPUT "FAIRE 1 POUR TIRER
, 0 POUR FINIR";R
70 IF R=0 THEN 140
80 GOSUB 410
90 T=T+X
100 PAPER 3:PEN 2
110 LOCATE 10,12:PRINT USING"VOTRE TOTAL
 EST DE ##";T
120 PAPER 0:PEN 1
130 GOTO 60
140 CLS
150 PRINT "LA BANQUE JOUE"
160 S=0
170 GOSUB 410
180 S=S+X
190 PAPER 1:PEN 0
200 PRINT USING "TOTAL DE LA BANQUE ##";
S
210 PAPER 0:PEN 1
220 FOR I=1 TO 500:NEXT I
230 IF S>T OR S>17 THEN 250
240 GOTO 170
250 CLS
260 PAPER 3:PEN 2:PRINT USING "VOUS AVEZ
   ##";T
270 PRINT
280 PAPER 1:PEN 0:PRINT USING "LA BANQUE
 A ##";S
290 PAPER 0:PEN 1
300 PRINT
310 IF S>21 OR (S<T AND T<=21) THEN 330
320 M=-M
330 PRINT "GAIN ";M
340 PRINT
350 G=G+M
360 PRINT "VOTRE TOTAL ACTUEL ";G
370 PRINT
380 INPUT "ON CONTINUE (O/N):";R$
390 IF R$="N" THEN END
400 GOTO 30
410 X=INT(6*RND)+1
420 RETURN
```

Pair-impair

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
C'est un jeu de hasard, pour un ou plusieurs joueurs. AMSTRAD lance simultanément deux dés, non visibles et pose trois questions : quelle est la parité du total, quel est le total, et enfin donner la valeur de l'un des dés.
On peut ensuite soit poursuivre le jeu, soit l'interrompre à tout moment.

LE PROGRAMME :
Les dés sont tirés par un sous-programme placé aux lignes 38Ø et 39Ø.
La ligne 8Ø tient le compte du nombre de parties, et les lignes 18Ø, 25Ø et 29Ø comptabilisent les points marqués.
La parité du total est déterminée par les lignes 13Ø et 14Ø (la ligne 14Ø compare avec la réponse du joueur).
La ligne 3ØØ et les suivantes réalisent l'affichage de fin de partie : total, nombre de points, nombre de parties jouées.
La ligne 36Ø provoque l'arrêt si l'on répond N à la question ON CONTINUE.

POUR JOUER :
On lance le programme par l'instruction RUN.
Il suffit ensuite de répondre aux diverses questions posées, en suivant scrupuleusement les indications données et en tapant ENTER après chaque réponse.
Lorsque l'on a terminé une partie, on obtient l'affichage du total actuel et du nombre de parties sous la forme de, par exemple, VOTRE TOTAL 6 EN 1Ø PARTIES.
On obtient alors l'affichage de la question ON CONTINUE (O/N) et l'on devra répondre N suivi de ENTER pour arrêter.

EXTENSIONS POSSIBLES :
Si l'on désire jouer à plusieurs, il sera intéressant de prévoir la possibilité de miser, et d'ajouter une gestion du score réalisé par chacun des joueurs.
On pourra aussi prévoir une amélioration sur le plan de la présentation (utilisation de la couleur, etc.).

```
10 REM PAIR-IMPAIR
20 RANDOMIZE (TIME)
30 INK 3,3,6
40 T=0:R=0
50 GOSUB 380
60 A=X
70 GOSUB 380
80 R=R+1
90 S=A+X
100 CLS
110 INPUT "TOTAL PAIR (0) OU IMPAIR (1)
:";N
120 PRINT
130 P=S/2-INT(S/2)
140 IF N=P OR N-0.5=P THEN 170
150 PEN 3:PRINT "FAUX":PEN 1
160 GOTO 190
170 PEN 2:PRINT "EXACT":PEN 1
180 T=T+1
190 PRINT
200 INPUT "TOTAL PRESUME :";N:PRINT
210 IF N=S THEN 250
220 PEN 3:PRINT "PERDU":PEN 1
230 FOR I=1 TO 1500:NEXT I
240 GOTO 310
250 T=T+2
260 PEN 2:PRINT "BRAVO! ";:PEN 1
270 INPUT "DONNEZ UN DE :";N:PRINT
280 IF N<>A AND N<>S-A THEN 220
290 T=T+3
300 PEN 2:PRINT "BRAVO!":PEN 1:GOTO 230
310 CLS
320 PRINT "VOTRE TOTAL:";T;"EN";R;"PARTI
E";
330 IF R>1 THEN PRINT "S"
340 PRINT
350 INPUT "ON CONTINUE (O/N)";R$
360 IF R$="N" THEN END
370 GOTO 50
380 X=INT(RND*6)+1
390 RETURN
```

Devine un chiffre

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
C'est un jeu de hasard, de conception très simple, mais néanmoins assez passionnant : le joueur introduit sa mise, et parie sur un chiffre, de 1 à 6 ; trois dés sont alors lancés ; si le chiffre joué apparaît deux fois il double sa mise ; s'il apparaît trois fois, il triple sa mise.

LE PROGRAMME :
Le tirage des dés est réalisé par un sous-programme placé en lignes 1000 et 1010.
L'affichage est réalisé par les lignes 150 et 160, la ligne 145 ayant déplacé les tabulations.
Les lignes 180 à 210 calculent le gain dans les diverses configurations possibles.
La totalisation est effectuée en ligne 230 pour les gains, et en ligne 510 pour les pertes.
La variable R$, testée en ligne 550, détermine la poursuite éventuelle du jeu.

POUR JOUER :
Le lancement du programme par RUN provoque un premier affichage VOTRE MISE ; on répond par un nombre suivi de ENTER. On obtient alors un deuxième affichage DONNEZ UN CHIFFRE DE 1 A 6 ; on entre un chiffre suivi de ENTER ; les 3 dés, repérés par les lettres A, B, C s'affichent alors, ainsi qu'un message indiquant votre gain ou votre perte. Puis une indication sur votre total actuel apparaît, ainsi que la question ON CONTINUE (O/N) ; taper O pour continuer, N pour arrêter (suivi de ENTER).

EXTENSIONS POSSIBLES :
On pourra facilement introduire des contrôles (mise minimum, coup proposé de 1 à 6, etc.). On pourra aussi dessiner des dés avec un graphisme plus élaboré, introduire la possibilité de jouer à plusieurs, faire même participer AMSTRAD.

```
10 REM DEVINE UN CHIFFRE
20 RANDOMIZE (TIME)
40 G=0
50 CLS
60 INPUT "VOTRE MISE: ",N
70 PRINT
80 PRINT "DONNEZ UN CHIFFRE DE 1 A 6: ";
:PEN 2:INPUT"",D:PEN 1
90 GOSUB 1000
100 A=X
110 GOSUB 1000
120 B=X
130 GOSUB 1000
140 C=X
145 ZONE 10
150 PRINT:PRINT:PRINT ," A"," B"," C"
160 PRINT:PEN 2:PRINT ,A,B,C:PEN 1
170 T=0
180 IF D=A THEN T=T+N
190 IF D=B THEN T=T+N
200 IF D=C THEN T=T+N
210 IF T=0 THEN 500
220 PRINT:PRINT:PRINT"VOUS GAGNEZ";T
230 G=G+T
240 GOTO 520
500 PEN 3: PRINT:PRINT:PRINT"PERDU":PEN
1
510 G=G-N
520 PRINT:PRINT:PRINT"TOTAL ACTUEL :";G
530 PRINT:PRINT
540 INPUT"ON CONTINUE (O/N)";R$
550 IF R$="N" OR R$="n" THEN END
560 GOTO 50
1000 X=INT(RND*6)+1
1010 RETURN
```

Le jeu de Wytthof

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
On dispose d'une grille carrée de 1Ø cases sur 1Ø cases. Un point apparaît, en dehors de la diagonale et les joueurs devront le déplacer à chaque tour, suivant seulement trois directions : droite, bas et diagonale, bas et droite. Le premier joueur qui arrive en (9,9) sera considéré comme gagnant, ou perdant, selon la convention faite au début.

LE PROGRAMME :
Les lignes 4Ø, 5Ø et 6Ø permettent de définir une position de départ, en dehors de la diagonale.
La ligne 8Ø et la boucle I (9Ø-11Ø) dessinent le terrain de jeu.
La ligne 13Ø imprime la position du point mobile, (une face qui devient souriante à la case d'arrivée (9,9).
Le test de fin est effectué en ligne 14Ø.
Le changement de joueur est assuré par la ligne 2ØØ.
Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 22Ø, 24Ø et 25Ø.

POUR JOUER :
Taper RUN pour lancer l'exécution du programme. Le terrain de jeu s'affiche, ainsi que la position de départ.
Puis on obtient l'affichage JOUEUR 1 et en dessous la question DIRECTION (1, 2, 3), à laquelle on répond en tapant l'un de ces trois nombres, suivi de ENTER.
C'est ensuite au tour du joueur 2.
Le jeu se termine lorsque l'un des joueurs arrive en case (9, 9), faire alors ESC deux fois pour reprendre la main.

EXTENSIONS POSSIBLES :
On peut tout d'abord améliorer les contrôles (la sortie du terrain n'est pas interdite, par exemple), améliorer ensuite l'affichage (graphisme, couleur), étendre le jeu à plusieurs joueurs avec un terrain agrandi, introduire un monstre qui va essayer de capturer un des joueurs, prévoir un programme qui permettrait à votre AMSTRAD de jouer, etc.

```
10 REM JEU DE WYTTHOF
20 RANDOMIZE (TIME)
30 DEFINT X,Y
40 X=RND*5
50 Y=RND*5
60 IF Y=X THEN 50
70 CLS
80 PRINT "      0  1  2  3  4  5  6  7  8
  9"
90 FOR I=0 TO 9
100 PRINT:PRINT " ";I;" .  .  .  .  .  .
  .  .  .  ."
110 NEXT I
120 H=1
130 LOCATE 3*X+6,2*Y+3:PRINT CHR$(225+(X
=9 AND Y=9))
140 IF X=9 AND Y=9 THEN PEN 1:GOTO 140
150 PEN H+1
160 LOCATE 1,23:PRINT STRING$(30,32)
170 LOCATE 1,23:PRINT "JOUEUR";H;:INPUT"
DIRECTION (1,2,3)";D$
180 D=VAL(D$)
190 ON D GOSUB 220,240,250
200 H=2-ABS(H-1)
210 GOTO 130
220 X=X+1
230 RETURN
240 GOSUB 220
250 Y=Y+1
260 RETURN
```

Rebonds

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu d'adresse.
Difficulté du jeu : moyenne.

LE JEU :
Il utilise l'idée de la simulation d'un billard à bandes.
De part et d'autre de l'écran se trouvent la balle et le but visé : une série de chiffres et de lettres, en haut de l'écran, permettent de choisir l'endroit où l'on désire rebondir.
La balle va alors partir vers le haut de l'écran, y rebondir, revenir éventuellement frapper le bas, rebondir, et ainsi de suite jusqu'à atteindre la ligne de but ; si l'on est suffisamment près de la cible on a gagné et le nombre de rebonds sera alors affiché.

LE PROGRAMME :
La ligne 6Ø imprime les chiffres et les lettres en haut de l'écran.
Les lignes 7Ø et 8Ø définissent les positions initiales de la balle et de la cible.
Les lignes 1ØØ et 11Ø effectuent leur impression.
Le symbole correspondant à la direction choisie est entré en ligne 12Ø.
La boucle J (18Ø-22Ø) réalise le déplacement de la balle.
La ligne 21Ø imprime la nouvelle position.
Le changement de direction, lors de la rencontre d'un bord, est effectué par la ligne 22Ø.
La ligne 24Ø effectue l'impression de fin de partie.
Le nombre de rebonds est compté en ligne 2ØØ.

POUR JOUER :
On lance le programme par RUN.
Le terrain de jeu est affiché, ainsi que les positions initiales de la balle et de la cible. On choisit l'endroit où l'on désire faire rebondir la balle (de 1 à Z), et on l'entre.
La trajectoire est alors affichée, suivie s'il y a lieu de la mention BRAVO ; X REBONDS.
On peut lancer une autre partie en tapant une touche quelconque, et s'arrêter à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
Il sera très intéressant, en utilisant un graphisme plus fin, de dessiner un véritable billard, avec son tapis vert, et d'indiquer la direction par un segment, symbolisant la queue de billard, se déplaçant autour de la bille.
On peut aussi ajouter des trous, mettre plusieurs billes, sonoriser les rebonds, ou les heurts avec d'autres balles, les comptabiliser pour faire une partie à plusieurs joueurs, etc.

```
10 REM REBONDS
20 RANDOMIZE (TIME)
30 DEFINT X,Y
40 INK 0,9:INK 3,3
50 CLS
60 PRINT "  0123456789ABCDEFGHIJKLMNOPQR
STUVWXYZ"
70 X=4+RND*18
80 Y=2+RND*18
90 PEN 3
100 LOCATE 1,X:PRINT CHR$(164)
110 LOCATE 40,Y: PRINT "J"
120 PEN 1:LOCATE 1,25:INPUT "VOTRE COUP:
",R$:PEN 3
130 R$=UPPER$(R$)
140 V=3+ASC(R$)-ASC("0")+7*(R$>="A")
150 S=0
160 T=X
170 DT=(2-X)/(V-1)
180 FOR J=2 TO 40
190 T=T+DT
200 IF T<2 OR T>25 THEN DT=-DT:T=25+23*(
DT>0):S=S+1
210 LOCATE J,T:PRINT CHR$(164);
220 NEXT J
230 LOCATE 1,12
240 IF ABS (T-Y)<2 THEN PRINT "BRAVO!";S
;"REBOND";STRING$(-(S>1),"S")
250 INPUT "UNE AUTRE PARTIE (O/N)";R$:PE
N 1
260 IF R$="O" THEN RUN
```

Jackpot

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : hasard (un joueur).
Difficulté du jeu : facile.

LE JEU :
C'est une variante simplifiée d'un jeu de hasard classique, dans lequel le joueur doit faire une mise. S'il obtient certaines combinaisons de symboles, il gagne (plus ou moins, selon sa mise et la combinaison obtenue).
Ici, les symboles sont remplacés par des chiffres, et le fait d'avoir trois chiffres identiques permet de gagner. Il existe de plus une combinaison particulière qui permet de remporter le Jackpot.

LE PROGRAMME :
La boucle I (7Ø-9Ø) permet le dessin de la machine, la boucle I (11Ø-13Ø) affichant les emplacements (vides) des chiffres.
La mise sera entrée en ligne 14Ø, et testée en ligne 15Ø : si on mise Ø, la partie est terminée (départ vers la ligne 35Ø).
Les lignes 16Ø, 18Ø et 2ØØ permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 33Ø.
Le résultat est affiché par la ligne 22Ø, puis testé en ligne 24Ø, et en ligne 31Ø, si l'on a gagné.

POUR JOUER :
Lancer le programme par RUN ; à l'affichage de la question POT DE DEBUT, on répond en entrant un nombre suffisamment élevé, sur lequel vos mises seront décomptées.
Puis on tape ENTER, et on entre une première mise (différente de Ø), suivie de ENTER.
Les trois chiffres tirés sont alors affichés un bref instant, ainsi que la mise faite, et le résultat obtenu (gain ou perte).
On peut ensuite miser de nouveau, faire ENTER et ainsi de suite.
Pour terminer, il suffit de répondre Ø à la question VOTRE MISE.
Le restant de votre pot initial sera alors affiché.

EXTENSIONS POSSIBLES :
On pourra envisager d'améliorer sérieusement les contrôles, utiliser la couleur et des possibilités graphiques plus élaborées pour simuler davantage un véritable Jackpot.

```
10 REM JACKPOT
20 DEFINT X
30 RANDOMIZE (TIME)
40 INK 0,9:BORDER 9
50 INK 1,1:INK 2,16,23:INK 3,3
60 CLS
70 FOR I=1 TO 5
80 LOCATE 10,I+4:PAPER 3:PRINT STRING$(2
1,32)
90 NEXT I
100 PAPER 0
110 FOR I=1 TO 3
120 LOCATE 14,5+I:PRINT "    ":LOCATE 19
,5+I:PRINT "    ":LOCATE 24,5+I:PRINT "
   "
130 NEXT I
140 LOCATE 1,14:INPUT"VOTRE MISE:";M
150 IF M=0 THEN 350
160 GOSUB 330
170 A=X
180 GOSUB 330
190 B=X
200 GOSUB 330
210 PEN 2
220 LOCATE 14,7:PRINT A:LOCATE 19,7:PRIN
T B:LOCATE 24,7:PRINT X
230 PEN 1
240 IF A=B AND A=X THEN 300
250 N=-M
260 LOCATE 7,14:PRINT USING "MISE :####
 !  RESULTAT :#####";M,N
270 LOCATE 4,25:INPUT "TAPER <ENTER> POU
R CONTINUER";R$
280 P=P+N
290 GOTO 60
300 N=M*50
310 IF A=7 THEN N=M*500
320 GOTO 260
330 X=9*RND+1
340 RETURN
350 CLS
360 PRINT "VOUS AVEZ ";P;"A BIENTOT!"
```

Roulette

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

LE JEU :
Il est surtout spectaculaire par son aspect graphique : une boule, représentée par une bille, se déplace sur un terrain de jeu comportant des cases numérotées de 1 à 36, ainsi que trois cases supplémentaires, le ØØ, une case P (comme pair) et une case I (comme impair).
La boule parcourt au moins une fois chaque case, puis s'arrête au deuxième tour, on peut ensuite la relancer pour une autre partie.

LE PROGRAMME :
La boucle K (6Ø-12Ø) dessine le terrain de jeu, auquel la ligne 13Ø rajoute les trois symboles P, ØØ et I.
La boucle K (15Ø-23Ø) réalise le déplacement de la boule, l'arrêt étant contrôlé par la ligne 2ØØ.
La boucle J (ligne 21Ø) réalise une légère temporisation variable pour simuler le ralentissement de la balle.
La ligne 21Ø efface la bille en déplaçant le curseur d'impression vers la gauche.
La ligne 24Ø tire la position d'arrêt.
La ligne 26Ø gère les essais suivants.

POUR JOUER :
Il suffit de faire RUN, et de regarder.
Une fois la boule arrêtée, il suffira de frapper une touche quelconque (sauf "N") pour la relancer. Taper N pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra essayer de se rapprocher davantage d'une véritable roulette, de différentes manières : mieux dessiner le terrain de jeu, introduire des couleurs, simuler le ralentissement de la boule, introduire différentes possibilités de pari, et assurer la gestion des gains correspondants, tenir les comptes de plusieurs joueurs, et annoncer les résultats sous une forme plus usuelle.

```
10 REM ROULETTE
20 RANDOMIZE(TIME)
30 DEFINT X
40 INK 0,9:BORDER 9:INK 2,26
50 CLS
60 FOR K=1 TO 4
70 FOR I=1 TO 9
80 S=I+I
90 PRINT TAB(4*I) (S AND K=1)+(S+18 AND
K=2)+(S-1 AND K=3)+(S+17 AND K=4);
100 NEXT I
110 PRINT:PRINT:PRINT:PRINT
120 NEXT K
130 LOCATE 1,3:PRINT "P":LOCATE 1,7:PRIN
T "OO":LOCATE 1,11:PRINT"I"
140 PEN 2
150 FOR K=1 TO 4
160 FOR I=1 TO 10+(K=4)
170 IF I=10 AND K<4  THEN LOCATE 1,4*K E
LSE LOCATE 4*I+1,4*K-1
180 PRINT CHR$(231);
190 X=X-1
200 IF X=0 THEN GOTO 260
210 FOR J=1 TO 30*(1-(39/X)*(X>0)):NEXT
J:PRINT CHR$(8);" "
220 NEXT I
230 NEXT K
240 X=1+RND*39
250 GOTO 150
260 PEN 1:PAPER 3:LOCATE 1,20:INPUT "AUT
RE ESSAI (O/N):";R$:PAPER 0
270 IF R$="N" THEN END
280 GOTO 50
```

Couloirs

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il fait appel à des notions de logique élémentaire. Le joueur peut se présenter à l'entrée, soit du couloir numéro 1, soit du couloir numéro 2. Dans ces deux entrées sont disposés des tourniquets qui basculent à votre passage et vous laissent passer, une fois vers la droite, et la fois suivante vers la gauche. De plus, un tourniquet analogue a été disposé dans le couloir central, pourrez-vous prévoir où (sortie A ou B) va s'effectuer votre prochaine sortie ?

LE PROGRAMME :
Les lignes 3Ø à 14Ø dessinent les couloirs (et les trois tourniquets) à l'aide du sous-programme 36Ø qui déplace le curseur d'impression (codes ASCII 8 et 1Ø).
La ligne 18Ø attend votre choix.
Le décodage du chemin est fait par AMSTRAD en utilisant trois variables logiques I, J et K, représentant les états des trois tourniquets (lignes 21Ø à 35Ø).

POUR JOUER :
Après la frappe de RUN, le couloir est dessiné sur l'écran, les tourniquets étant visibles, et l'on devra taper 1 ou 2 en réponse à la question de l'AMSTRAD.
Dans la version retenue ici, la réponse est alors immédiatement affichée, et le chemin suivi depuis le début du jeu, ainsi que les sorties correspondantes, sont affichés sur l'écran. Ceci présente l'avantage de permettre à plusieurs joueurs d'exercer leur sagacité en essayant de prévoir, pour une entrée donnée, ce que va répondre AMSTRAD.

EXTENSIONS POSSIBLES :
Elles seront très nombreuses ; il est très facile d'imaginer un couloir plus complexe avec d'autres entrées et d'autres sorties, et d'autres tourniquets. On peut imaginer un affichage graphique des résultats en dessinant, un bref instant, l'état des tourniquets après la réponse du joueur. On peut aussi ajouter un test des réponses (jeu à plusieurs), décompter les points de chaque joueur, etc.).

```
10 REM COULOIRS
20 CLS
30 PRINT STRING$(40,154)
40 PRINT
50 PRINT "B";SPACE$(6);STRING$(26,154);S
PACE$(6);"2"
60 PRINT
70 PRINT STRING$(14,154);SPACE$(12);STRI
NG$(14,154)
80 PRINT
90 PRINT "A";SPACE$(6);STRING$(26,154);S
PACE$(6);"1"
100 PRINT
110 PRINT STRING$(40,154)
120 PEN 3:LOCATE 33,3:GOSUB 360
130 LOCATE 14,6:GOSUB 360
140 LOCATE 33,9:GOSUB 360
150 I=0:J=0:K=0
160 R$="":T$=""
170 PEN 2
180 LOCATE 1,17:PRINT CHR$(18);:INPUT "E
NTREE 1 OU 2:";C$
190 IF C$="" OR C$>="3" OR C$<"1" OR LEN
 (C$)<>1 THEN 180
200 R$=R$+C$
210 PRINT:PRINT R$
220 IF C$="1" THEN 270
230 J=NOT J
240 IF J<>0 THEN 330
250 S$="B"
260 GOTO 300
270 I=NOT(I)
280 IF I<>0 THEN 330
290 S$="A"
300 T$=T$+S$
310 LOCATE 1,21:PRINT "SORTIE EN ";S$:PR
INT:PRINT T$
320 GOTO 180
330 K=NOT (K)
340 IF K<>0 THEN 250
350 GOTO 290
360 PRINT CHR$(205);" ";CHR$(204);
370 PRINT STRING$(2,8);CHR$(10);CHR$(203
);
380 PRINT STRING$(2,8);CHR$(10);CHR$(204
);" ";CHR$(205)
390 RETURN
```

Echiquier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : graphisme.
Difficulté du jeu : variable, selon utilisation.

LE JEU :
Le programme proposé vient en appoint du programme Echecs ; il donne, en effet, la possibilité de réaliser un affichage graphique de bonne qualité pour des jeux utilisant un damier de 8 cases (on pourra facilement l'étendre à 1Ø cases sur 1Ø cases). Un repérage a été prévu, horizontalement, par des lettres (de A à H), et, verticalement, par des nombres (de 1 à 8).

LE PROGRAMME :
Les lignes 5Ø, 6Ø et 7Ø définissent trois variables A$, B$ et C$ qui contiennent les éléments de dessin. La ligne 8Ø imprime les lettres horizontales sur une fenêtre, ayant sa propre couleur d'impression.
La boucle I (9Ø-15Ø) permet l'impression, le motif à imprimer étant affiché ou non selon la valeur de (I + J)/2 ; la ligne 12Ø réalise l'impression.

POUR JOUER :
L'ordre RUN provoque l'affichage du damier sur l'écran.
Pour utiliser le programme, il conviendra de tenir compte du repérage des cases utilisables qui sont les centres des carrés noirs ou gris. Les cases sont obtenues en largeur par une tabulation de la forme 3*J, J variant de 1 à 8, et en hauteur par 2*I.

EXTENSIONS POSSIBLES :
On pourra utiliser ce programme pour toute sorte de jeux se déroulant sur un damier (échecs, dames anglaises, le loup et les agneaux, etc.).

```
10 REM ECHIQUIER
20 CLS
30 INK 0,0:BORDER 0:INK 1,26:INK 2,2
40 PEN #1,2:LOCATE #1,1,1
50 A$=STRING$(3,140)
60 B$=STRING$(3,143)
70 C$=STRING$(3,131)
80 PRINT #1,"     A  B  C  D  E  F  G  H"
:PRINT #1
90 FOR I=1 TO 8
100 FOR J=1 TO 8
110 IF (I+J)/2=INT((I+J)/2) THEN 130
120 LOCATE 3*J+1,2*I:PRINT A$:LOCATE 3*J
+1,2*I+1:PRINT B$:LOCATE 3*J+1,2*I+2:PRI
NT C$
130 NEXT J
140 PRINT #1,I:PRINT #1
150 NEXT I
160 GOTO 160
```

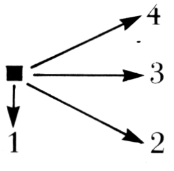
Le jeu de la fève

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Présentant une certaine analogie avec le jeu de WYTTHOF, cette version offre en fait 4 possibilités de déplacement ; la position de départ est ici fixée en (Ø, Ø), et pourrait éventuellement être modifiée.
Les directions de déplacement disponibles sont : droite, bas, diagonale bas et droite, diagonale haut et droite.
On joue en principe avec la règle : le premier arrivé en (9, 9) a gagné.

LE PROGRAMME :
La ligne 6Ø et la boucle I (7Ø-9Ø) dessinent le terrain de jeu.
La ligne 13Ø réalise le test de fin de partie.
La ligne 18Ø permet d'exécuter l'un des sous-programmes de déplacement.
Chaque joueur dispose d'une fenêtre d'impression avec une couleur, qui est superposée au dessin du terrain de jeu.
La ligne 14Ø assure le changement de joueur.

POUR JOUER :
On lance le programme par RUN, et le terrain de jeu s'affiche, ainsi que la question DIRECTION (1, 2, 3, 4) ; les 4 directions proposées correspondent aux déplacements suivants :



Le joueur 1 entre son déplacement, et tape ENTER.
Le mouvement correspondant est réalisé, puis on obtient l'affichage JOUEUR 2 et DIRECTION.
Le joueur 2 procède de même, et ainsi de suite, jusqu'à ce que l'un des joueurs arrive en (9, 9) auquel cas le jeu se termine.

EXTENSIONS POSSIBLES :
On peut imaginer d'introduire en début de partie des cases pièges (chacun essayant alors de faire venir l'autre sur l'une de ces cases).
On peut aussi jouer avec plusieurs symboles, si possible de couleurs différentes, déplacés simultanément, envisager des terrains de jeu triangulaires, aux dimensions arbitraires, etc.

```
10 REM JEU DE LA FEVE
20 DEFINT I,X,Y
30 CLS
40 INK 0,13:INK 1,9:INK 2,15:INK 3,3

50 PEN #1,2:PEN #2,1
60 PEN 3:PRINT "     0  1  2  3  4  5  6
  7  8  9"
70 FOR I=1 TO 9
80 PRINT:PRINT I; "  .  .  .  .  .  .  .
  .  .  ."
90 NEXT I
100 X=0:Y=1
110 J=2
120 LOCATE #J,3*X+6,2*Y+1:PRINT #J,CHR$(
248+J)
130 IF X=9 AND Y=9 THEN 200
140 J=1+J MOD 2
150 LOCATE #J,1,23:PRINT CHR$(20);
160 PRINT #J,"JOUEUR";J;"- DIRECTION (1"
;CHR$(241);",2";CHR$(205);",3";CHR$(243)
;",4";CHR$(204);"):";:INPUT #J,"",I$
170 I=VAL (I$)
180 ON I GOSUB 220,240,250,270
190 GOTO 120
200 LOCATE #J,1,23:PRINT #J,"LE JOUEUR";
J;"A GAGNE....                 "
210 GOTO 210
220 Y=Y+1
230 RETURN
240 GOSUB 220
250 X=X+1
260 RETURN
270 GOSUB 250
280 Y=Y-1
290 RETURN
```

Jeu de l'oie aléatoire

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard (deux joueurs).
Difficulté du jeu : facile.

LE JEU :
Le jeu reprend la très ancienne idée du jeu de l'oie, mais en y ajoutant la possibilité offerte par l'ordinateur de choisir au hasard les cases "à problème".
La règle du jeu est sans grand intérêt, puisque les joueurs n'interviennent à aucun moment dans le déroulement de la partie, qui est entièrement gérée par AMSTRAD ; il suffira d'appuyer sur une touche après chaque lancer de dé, et de regarder ce qui se passe.

LE PROGRAMME :
La ligne 8Ø permet d'appeler le sous-programme 28Ø à 38Ø qui réalise le contrôle de la position des joueurs et l'affichage de cette position. Cet affichage sera d'ailleurs refait en cours de partie, pour tenir compte du déplacement réellement effectué (qui n'est pas nécessairement celui indiqué par le dé).
La ligne 9Ø s'occupe du changement de joueur, et la ligne 1ØØ effectue le lancer de dé.
La ligne 37Ø traite le cas où l'un des joueurs termine.
La ligne 14Ø tire un nombre aléatoire entre Ø et 3, qui servira s'il y a lieu pour déterminer le vrai déplacement du joueur, en envoyant dans l'un des sous-programmes de "chance".
Les lignes 28Ø et 29Ø font avancer les pions, et les lignes 3ØØ à 35Ø redessinent à chaque affichage le terrain de jeu en entier.

POUR JOUER :
On lancera le jeu par RUN. Le terrain de jeu est alors dessiné, puis on voit apparaître l'affichage JOUEUR 1 :...
On devra alors appuyer sur une touche quelconque pour que le coup du joueur numéro 1 soit pris en compte ; ne pas s'inquiéter du temps de réaction, qui pourra sembler un peu long.
Le déplacement indiqué par les dés sera exécuté, puis au bout de quelque temps, un second déplacement sera éventuellement effectué : c'est le déplacement "CHANCE". On obtiendra alors l'affichage JOUEUR 2 :...
Le joueur 2 appuie à son tour sur une touche quelconque et ainsi de suite...
Le joueur gagnant sera signalé par un affichage BRAVO en fin de partie.
On peut aussi s'arrêter à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
On peut agrandir le terrain de jeu, introduire davantage d'aléas, changer leur nature. On pourrait aussi choisir les cases "**CHANCE**" en début de jeu, ajouter des affichages en conséquence, etc.

```
10 REM JEU DE L'OIE ALEATOIRE
20 DEFINT I,J,X,Y
30 RANDOMIZE (TIME)
40 INK 0,3:INK 1,8:INK 2,18:INK 3,11
50 CLS
60 J=2
70 LOCATE 1,3:PRINT "  0123456789ABCDEFG
HIJKLMNOPQRSTUVWXYZ"
80 GOSUB 280
90 J=1+J MOD 2
100 I=1+RND*5
110 PEN J+1:PRINT CHR$(7);:LOCATE 8,14:P
RINT "JOUEUR";J;"- VOTRE COUP:";I
120 PRINT TAB(10) "APPUYER SUR <ENTER>";
CHR$(20);:INPUT "",R$
130 GOSUB 280
140 I=RND*3
150 F=1+RND*4
160 ON F GOSUB 180,200,220,260,250
170 GOTO 80
180 I=I+3
190 RETURN
200 I=-I
210 RETURN
220 I=2*I
230 RETURN
240 I=0
250 RETURN
260 IF J=1 THEN X=RND*30 ELSE Y=RND*30
270 RETURN
280 IF J=1 THEN X=X+I*((X+I)<=35)*((X+I)
>=0)
290 IF J=2 THEN Y=Y+I*((Y+I)<=35)*((Y+I)
>=0)
300 FOR I=0 TO 35
310 LOCATE 3+I,5:PRINT " ";
320 IF I=X THEN LOCATE 3+I,5:PEN 2:PRINT
 "1";
330 LOCATE 3+I,7:PRINT " ";
340 IF I=Y THEN LOCATE 3+I,7:PEN 3:PRINT
 "2";
350 NEXT I
360 FOR I=1 TO 600:NEXT I
370 IF X=35 OR Y=35 THEN LOCATE 1,20:PRI
NT"BRAVO":GOSUB 390:END
380 RETURN
390 SOUND 1,478: SOUND 1,319: SOUND 1,379
400 RETURN
```

Niveau 2

Nom du programme	*Thèmes et particularités*
Dé truqué	A utiliser, si vous êtes malhonnête, dans d'autres programmes.
Un dé	Tirage d'un dé, avec graphisme.
Visions	Un jeu sur les mots, pour l'ordinateur.
421	Une simulation simplifiée, plusieurs joueurs.
L'électricien fou	Logique et difficile, pour un seul joueur.
La course au 20	Jeu de réflexion, pour deux joueurs ou plus.
Musique	Révisez vos notions de solfège, grâce à l'ordinateur.
Poker	Une base de départ, pour mettre au point votre programme.
Le jeu de saute-mouton	Un jeu de réflexion, pour deux joueurs.
Lettres	Un jeu sur les mots, assez simple.
Machinamot	Laissez l'ordinateur créer des vocables nouveaux.
Espion	Dites-lui votre amour en code.
Jeu de loto	Gagnez sans problème, au prochain tirage.
Tirage d'un dé truqué	Révisez vos connaissances en statistiques.
Le jeu de Marienbad	Un jeu de logique pour deux joueurs, dans une version étendue.
Compose	Faire de la musique, c'est d'abord l'écrire...
Jour de la semaine	Profitez-en pour faire le point sur les anniversaires.
Tour de cartes	Votre ordinateur a plus d'un tour dans son sac.
L'indésirable	Vous êtes propriétaire. Arriverez-vous à mettre la main sur ce locataire indésirable ?
Le jeu de la vie	Un grand classique, en deux versions : manuelle et automatique.

Dimensionnement d'un tableau Recherche d'un élément dans un tableau et dans une chaîne L'instruction DATA

L'INSTRUCTION DIM OU INSTRUCTION DE DIMENSIONNEMENT D'UN TABLEAU

Les structures de tableaux sont très couramment utilisées en programmation, se révélant même indispensables dans un grand nombre de cas. Si vous voulez, par exemple, entrer vos recettes et dépenses pour tous les jours du mois (on supposera pour simplifier des mois de trente jours) vous pourriez définir 30 variables appelées PREMIER JOUR, DEUXIEME JOUR, etc. Il est plus facile, et plus rationnel d'utiliser une boucle I, I variant de 1 à 3Ø, et une variable R (I) pour les recettes, et D (I) pour les dépenses ; ce faisant, vous avez, sans le savoir, utilisé une structure de tableau à une dimension ; ceci ne pourra être admis sur AMSTRAD que si vous avez auparavant prévenu que vous alliez utiliser un tel tableau, en donnant une instruction DIM R (3Ø) pour les recettes, et DIM D (3Ø) pour les dépenses.

On pourrait aussi imaginer de conserver dépenses et recettes dans un même tableau, en lui donnant cette fois deux dimensions : par exemple DIM RD (3Ø, 2) (ou DIM RD (2, 3Ø)). Ainsi, avec DIM RD (3Ø, 2) on pourra convenir que RD (I, 1) représente les recettes, et RD (I, 2) les dépenses. Des tableaux de plus de 2 dimensions sont également possibles.

On peut aussi concevoir des tableaux dont les éléments sont des chaînes de caractères, et non plus des nombres, avec 255 caractères au maximum.

Ainsi DIM NOM$ (12) réserve 12 emplacements en mémoire, pour des mots de 255 caractères au maximum.

Sur AMSTRAD il n'est pas obligatoire de dimensionner un tableau qui contient moins de 10 éléments.

LA RECHERCHE D'UN ELEMENT DANS UN TABLEAU : R (I)

Si l'on a un tableau à une dimension, numérique ou alphanumérique, défini par une instruction comme DIM R (3Ø) ou DIM N$ (2Ø), le Ième élément de ce tableau est repéré par R (I) (ou N$ (I)) ; il sera alors manipulé exactement comme une variable ordinaire (affectation, calcul, tests, etc.).

Ainsi INPUT R (3) permet d'entrer à la main la valeur du 3e élément numérique du tableau R. PRINT R$ (5) permet l'affichage du 5e élément du tableau alphanumérique R$, etc.

LA RECHERCHE D'UN ELEMENT DANS UNE CHAINE : MID$

Pour extraire le Ième élément de la chaîne alphanumérique A$, on utilisera l'instruction MID$.

Ainsi, si A$ = "BONJOUR", MID$ (A$, 3, 1) donne le caractère N ; de plus MID$ (A$, 3, 3) donne la sous-chaîne "NJO".

LE TRAITEMENT DE DONNEES MULTIPLES INSTRUCTIONS READ, DATA, RESTORE

Pour remplir facilement un tableau, numérique ou alphanumérique, il sera intéressant d'utiliser des lignes DATA, qui contiennent les données inscrites les unes à la suite des autres, séparées par une virgule.

Ainsi, on pourra écrire :

DATA LEON, JULES, 13, rue MARTIN...

Ces données peuvent être lues dans des boucles FOR... NEXT, au moyen de l'instruction READ. On peut recommencer la lecture au début des DATA, à l'aide de l'instruction RESTORE.

Dé truqué

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
On veut essayer de fabriquer un dé truqué selon sa propre convenance. La méthode proposée ici consiste à définir au départ une probabilité de réalisation pour chaque type de tirage (soit la probabilité d'obtenir un 1, un 2, un 3, etc. que chacune de ces probabilités soit comprise entre Ø et 1, et que la somme de toutes les probabilités soit un nombre très voisin de 1).

LE PROGRAMME :
Il utilise le générateur aléatoire d'AMSTRAD supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin est). La boucle I (6Ø-13Ø) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 1ØØ) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 8Ø du programme.
Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (18Ø-2ØØ), la ligne permettant le trucage se trouvant en 19Ø. Un nouveau tirage pourra être effectué sur pression d'une touche (ligne 22Ø).

POUR JOUER :
Après le lancement de l'exécution par RUN, on obtient l'affichage de la question PROBABILITE DE FAIRE 1, à laquelle on devra répondre par un nombre compris entre Ø et 1. Il conviendra de ne pas oublier que le total de toutes les probabilités entrées ne devra en aucun cas dépasser 1.1 (tolérance de Ø.1).
Des messages d'erreurs pourront être affichés et le programme sera éventuellement relancé. Lorsque toutes les probabilités auront été définies, le premier tirage sera automatiquement effectué, et il suffira de presser une touche quelconque pour effectuer les tirages suivants. On arrête le programme en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
Il sera évidemment bien tentant d'utiliser sournoisement ce petit programme pour améliorer vos chances de gain dans un jeu de hasard utilisant des lancers de dés. On pourra alors envisager de figer les données définissant le dé truqué dans une variable A$, etc. On pourra aussi envisager de présenter les résultats correspondant à une série de lancers de dés sous une forme spectaculaire, qui mette le trucage en évidence...

```
10 REM DE TRUQUE
20 DEFINT N
30 RANDOMIZE (TIME)
40 CLS
50 DIM E(6)
60 FOR I=1 TO 6
70 LOCATE 25,3:PRINT CHR$(18);
80 LOCATE 1,3:PRINT "PROBABILITE DE FAIR
E";I;:INPUT":",E(I)
90 P=P+E(I)
100 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN 24
0
110 LOCATE 6*I-3,10:PRINT "DE";I
120 LOCATE 6*I-3,12:PRINT E(I)
130 NEXT I
140 IF P<0.9 THEN PRINT CHR$(7);:RUN
150 LOCATE 1,3:PRINT CHR$(18);
160 FOR I=1 TO 500:NEXT I
170 N=1+RND*5
180 FOR I=1 TO 6
190 IF N=I AND RND>E(I) THEN 170
200 NEXT I
210 LOCATE 15,17:PAPER 2:PEN 3:PRINT "TI
RAGE :";N;:PAPER 0:PEN 1:GOSUB 290
220 LOCATE 3,25:INPUT "FAIRE <ENTER> POU
R UN AUTRE TIRAGE",R$
230 GOTO 170
240 LOCATE 10,25:PAPER 3:PRINT CHR$(7);"
IMPOSSIBLE: 0<=P<=1";:PAPER 0
250 P=P-E(I)
260 FOR J=1 TO 1500:NEXT J
270 LOCATE 10,25:PRINT STRING$(19,32);
280 GOTO 70
290 SOUND 1,478,20:SOUND 1,319,20:SOUND
1,379,20:RETURN
```

Un dé

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : tirage d'un dé (avec graphisme).
Difficulté du jeu : aucune.

LE JEU :
Le programme propose simplement le lancer d'un dé aléatoire, et sa représentation graphique en grand format.
Il pourra facilement être utilisé en sous-programme, dans tout type de jeux faisant appel au lancer d'un ou de plusieurs dés.

LE PROGRAMME :
La ligne 5Ø utilise le générateur aléatoire de l'AMSTRAD pour tirer un nombre de 1 à 6.
La boucle J (7Ø-9Ø) dessine les bords du dé.
Les lignes 12Ø à 17Ø sont utilisées pour le dessin des points.
Les lignes 18Ø et 19Ø permettront le cas échéant de refaire un tirage.

POUR JOUER :
Lancer le programme par RUN.
On entre un entier pour initialiser le générateur, et un dé s'affiche, en couleurs, et en grand format.
A la question : AUTRE TIRAGE (O/N) on répond O, pour relancer le dé.

EXTENSIONS POSSIBLES :
On pourra introduire une petite musique agrémentant chaque lancer, faire défiler les points comme si le dé roulait sur une table, etc.

```
10 REM UN DE
20 RANDOMIZE(TIME)
30 CLS
40 C=INT(RND*27):INK 1,C+(C=1)
50 A=INT(RND*6+1)
60 LOCATE 12,5:PRINT CHR$(150);STRING$(1
1,154);CHR$(156)
70 FOR J=0 TO 10
80 LOCATE 12,6+J:PRINT CHR$(149):LOCATE
24,6+J:PRINT CHR$(149)
90 NEXT J
100 LOCATE 12,17:PRINT CHR$(147);STRING$
(11,154);CHR$(153)
110 RESTORE 1000:FOR I=1 TO 3:READ N:SOU
ND 1,N:NEXT I
120 IF A<>1 THEN LOCATE 15,8:PRINT CHR$(
231)
130 IF A>=4 THEN LOCATE 21,8:PRINT CHR$(
231)
140 IF A=6 THEN LOCATE 15,11:PRINT CHR$(
231):LOCATE 21,11:PRINT CHR$(231)
150 IF A MOD 2 THEN LOCATE 18,11:PRINT C
HR$(231)
160 IF A>=4 THEN LOCATE 15,14:PRINT CHR$
(231)
170 IF A<>1 THEN LOCATE 21,14:PRINT CHR$
(231)
180 PEN 2:LOCATE 1,25:INPUT "AUTRE TIRAG
E (O/N)";R$
190 IF UPPER$(LEFT$(R$,1))="O" THEN PEN
1:GOTO 30
1000 DATA 478,319,379
```

Visions

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : facile.

LE JEU :
AMSTRAD vous permet, pour la première fois, d'entrer dans son intimité en vous laissant entrevoir sa propre représentation du monde ! Vous allez, en effet, lui proposer des mots, n'importe lesquels, et il va vous répondre en vous proposant une représentation graphique de ce que représente ce mot pour lui. Le jeu sera beaucoup plus amusant si vous le pratiquez à plusieurs, et vous pourrez quelquefois avoir des surprises.

LE PROGRAMME :
Une boucle I (40-120) se charge de réaliser les dessins.
La ligne 50 permet de transformer en nombre chaque lettre du mot entré au début (ligne 30). Ce nombre sera ensuite traduit en base 2 par une boucle J (lignes 60 à 110) ; selon le résultat de cette transformation, un point sera imprimé, ou non (ligne 90).

POUR JOUER :
On débute l'exécution du programme par RUN, ce qui amène l'affichage de la question QUEL DESSIN. On répond en entrant un mot quelconque qui sera transformé en dessin par AMSTRAD dès que l'on aura frappé ENTER.
Pour recommencer, il suffira de frapper une touche quelconque, et ESC deux fois pour finir.

EXTENSIONS POSSIBLES :
La même idée peut être reprise en introduisant des éléments de symétrie dans les dessins, en mettant plusieurs couleurs. On peut aussi imaginer de coder par cette méthode un certain nombre de dessins, que l'on introduirait au fur et à mesure dans la machine et l'on pourra alors créer un jeu dans lequel AMSTRAD deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs dans lequel AMSTRAD vous demande de trouver le mot qui correspond à un certain dessin, etc.

```
10 REM VISIONS
20 CLS
30 INPUT"QUEL DESSIN :";R$
40 FOR I=1 TO LEN(R$)
50 N=9*ASC(MID$(R$,I,1))-340
60 J=1
70 S=N/2-INT(N/2)
80 N=INT(N/2)
90 IF S<>0 THEN LOCATE J+15,I+5:PAPER 2:
PRINT " ":PAPER 0
100 J=J+1
110 IF J<10 THEN 70
120 NEXT I
130 LOCATE 11,25:INPUT "APPUYER SUR <ENT
ER>",R$
140 F=INT(RND*27):F=F+(F=1):INK 2,F
150 GOTO 20
```

421

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : moyenne.

LE JEU :
Chacun des joueurs devra, à son tour, lancer 3 dés, et peut ensuite choisir, sur les deux coups suivants, s'il désire relancer ou non 1, 2 ou 3 dés. Il marque alors un score correspondant au nombre indiqué par les dés, suivant les règles choisies, et le joueur suivant procède de la même manière.

LE PROGRAMME :
Le nombre de joueurs est tout d'abord entré en ligne 4Ø.
La ligne 13Ø affiche le numéro de joueur et le nombre d'essais déjà effectués.
Un sous-programme placé en ligne 4ØØØ, utilisé 3 fois dans la boucle J (2ØØ-25Ø), réalise le lancer de dés.
Les retirages sont effectués par les lignes 1ØØØ, 2ØØØ et 3ØØØ.
La variable R$ gère les 2e et 3e tirages. L'impression du résultat est réalisée dans tous les cas par les lignes 14Ø-15Ø.
Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 26Ø et 28Ø.

POUR JOUER :
On lance l'exécution du programme par RUN ce qui conduit à l'affichage NOMBRE DE JOUEURS. On entre le nombre de joueurs, et l'on tape ENTER. L'écran s'efface et affiche JOUEUR NUMERO 1. ESSAI K, PARTIE P, ainsi que le tirage réalisé, et les noms des dés (A, B, C).
Si l'on ne désire pas refaire de tirage, il suffit de frapper ENTER. La main passe alors au joueur suivant.
Pour effectuer un autre tirage, il suffit de taper le nom du ou des dés à retirer, et de faire ENTER : par exemple, pour retirer A et C on tapera AC suivi de ENTER.
Les joueurs suivants procéderont de la même façon.

EXTENSIONS POSSIBLES :
La plus évidente consiste à assurer la gestion des résultats des joueurs, en se référant aux règles de gain usuelles du jeu du 421. On pourra envisager l'affichage après chaque tour d'un tableau récapitulant les résultats des joueurs.
On pourra aussi améliorer l'affichage des dés (se référer aux divers programmes d'affichage des dés se trouvant dans le même ouvrage).

```
10 REM 421
20 RANDOMIZE(TIME)
30 INK 0,0
40 INPUT "NOMBRE DE JOUEURS :";N
50 P=1
60 I=1
70 K=0
80 CLS
90 R$="ABC"
100 GOTO 200
110 K=K+1
120 LOCATE 17,1:PEN 3:PAPER 1:PRINT "-42
1-":PAPER 0
130 INK 2,2+I MOD 26:PEN 2:LOCATE 1,4:PR
INT "JOUEUR NUMERO";I;"- ESSAI";K;"- PAR
TIE";P
140 LOCATE 7,7:PRINT A:LOCATE 17,7:PRINT
 B:LOCATE 27,7:PRINT C:PEN 1
150 LOCATE 8,9:PRINT"A":LOCATE 18,9:PRIN
T "B":LOCATE 28,9:PRINT "C"
160 R$="":LOCATE 7,14:PRINT CHR$(20);
170 LOCATE 1,14:INPUT "ABC :";R$
180 IF R$="" THEN 270
190 R$=R$+"DDD"
200 FOR J=1 TO 3
210 GOSUB 4000
220 X$=MID$(R$,J,1)
230 X=ASC(X$)-64:IF X<0 THEN X=4
240 ON X GOSUB 1000,2000,3000
250 NEXT J
260 IF K<3 THEN 110
270 I=I+1
280 IF I<=N THEN 70
290 P=P+1
300 GOTO 60
1000 A=D
1010 RETURN
2000 B=D
2010 RETURN
3000 C=D
3010 RETURN
4000 D=1+INT(RND*6)
4010 RETURN
```

L'électricien fou

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

LE JEU :
Un immeuble comportant 1Ø fenêtres, numérotées de Ø à 9, apparaît sur l'écran. Le joueur devra à chaque essai manœuvrer l'un des interrupteurs, en indiquant son numéro (de Ø à 9).
En principe, la fenêtre correspondante devrait s'éteindre, mais il est aussi possible qu'une autre s'allume (ou s'éteigne). Il s'agit, bien sûr, d'essayer d'obtenir l'extinction des 1Ø fenêtres, avec le moins possible d'essais.

LE PROGRAMME :
Une table A, en ligne 3Ø, contiendra l'état (allumé ou éteint) des 1Ø fenêtres.
La boucle I (5Ø-7Ø) et l'autre boucle I (9Ø-14Ø) dessinent l'immeuble dans son état initial.
La ligne 15Ø regarde si tout est éteint, auquel cas l'on ira en ligne 25Ø afficher le nombre d'essais.
La ligne 16Ø attend l'indication de l'interrupteur à manœuvrer.
Les lignes 17Ø à 23Ø définissent le résultat de la manœuvre.

POUR JOUER :
RUN provoque l'affichage de l'immeuble, avec toutes les fenêtres allumées.
A la question INTERRUPTEUR NUMERO, on répond en indiquant le numéro de l'interrupteur à manœuvrer, suivi de ENTER.
Le résultat de la manœuvre est alors affiché sur l'écran (des fenêtres s'allument ou s'éteignent).
On continue ainsi jusqu'à ce que toutes les fenêtres soient éteintes, et l'on obtient alors l'affichage du nombre d'essais réalisés.

EXTENSIONS POSSIBLES :
On pourra envisager un graphisme très amélioré avec si possible l'utilisation de la couleur. On pourra aussi définir un jeu à deux joueurs, augmenter le nombre de fenêtres, modifier la formule définissant l'allumage ou l'extinction...

```
10 REM L'ELECTRICIEN FOU
20 RANDOMIZE(TIME)
30 DIM A(10)
40 CLS
50 FOR I=1 TO 7
60 PAPER 2:LOCATE 10,I:PRINT STRING$(21,
32):PAPER 0
70 NEXT I
80 T=0
90 FOR I=1 TO 10
100 T=T+A(I)
110 LOCATE 2*I+9,4:PRINT " "
120 IF A(I)=0 THEN LOCATE 2*I+9,4:PRINT
CHR$(224)
130 PAPER 2:PEN 0:LOCATE 2*I+8,6:PRINT I
-1:PAPER 0:PEN 1
140 NEXT I
150 IF T=10 THEN 250
160 PEN 3:LOCATE 1,14:PRINT CHR$(7);CHR$
(20);:INPUT "INTERRUPTEUR NUMERO :";N:PE
N 1
170 N=N+1
180 M=M+1
190 A(N)=1-A(N)
200 R=TAN(RND+N/RND-N)-SIN(RND/N)+336*SI
N(8*N)
210 N=INT (10*(R-INT(R)))
220 IF N=0 THEN N=INT (RND*10+1)
230 IF A(N)=1 THEN A(N)=0:GOTO 200
240 A(N)=1:GOTO 80
250 LOCATE 1,14:PRINT "TROUVE EN";M;"ESS
AIS             "
```

La course au 20

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : logique (deux joueurs, ou plus).
Difficulté du jeu : moyenne.

LE JEU :
Il est possible d'imaginer pour ce jeu diverses règles, et nous n'avons retenu ici que la règle usuelle pour deux joueurs, qui en fait un jeu apparenté aux jeux de déplacement, ou d'allumettes : 4 tas de 2Ø étoiles vont apparaître en début de partie, et chaque joueur peut prendre autant d'étoiles qu'il le désire ; mais ceci dans un seul tas à la fois. Celui qui prend la dernière étoile est considéré comme perdant.

LE PROGRAMME :
La boucle I (3Ø-5Ø) dessine 4 rangées de 2Ø étoiles.
La variable F, dimensionnée en ligne 8Ø, servira à enregistrer le nombre d'étoiles prises dans chaque rangée.
La variable T contiendra le nombre total d'étoiles enlevées (ligne 7Ø).
La ligne 14Ø contrôle qu'il reste suffisamment d'étoiles sur la ligne indiquée.
La boucle J (15Ø-18Ø) supprime les étoiles ; la ligne 22Ø teste la fin de partie.

POUR JOUER :
Lancer le programme par RUN ; le terrain de jeu s'affiche, puis les inscriptions JOUEUR 1 et LIGNE, puis CASES. On répond en entrant un numéro de ligne (entre 1 et 4) suivi de ENTER, et le nombre d'étoiles à enlever suivi de ENTER. Les étoiles sont alors supprimées, et c'est au tour du deuxième joueur.
La partie se termine, lorsque la dernière étoile a été enlevée, par l'affichage du joueur gagnant.

EXTENSIONS POSSIBLES :
Outre une amélioration de la qualité du graphisme et des contrôles, de nombreuses variantes sont envisageables, en modifiant si besoin est, les règles de départ : on peut autoriser l'enlèvement d'un nombre impair d'étoiles seulement, ou alterner pair et impair, ou tirer au hasard le nombre à enlever, ou encore jouer à plusieurs...

```
10 REM COURSE AU 20
20 CLS:PEN 3
30 FOR I=1 TO 4
40 LOCATE 1,2*I:PRINT I;STRING$(20,238)
50 NEXT I
60 PEN 1
70 T=0
80 DIM F(4)
90 B=1
100 LOCATE 1,13:PEN B:PRINT CHR$(20);"JO
UEUR";B
110 PRINT:INPUT "LIGNE :";L
120 IF L<1 OR L>4 THEN 100
130 PRINT:INPUT "CASES :";C
140 IF C+F(L)>20 OR C<=0 THEN 100
150 FOR J=1 TO C
160 LOCATE 24-F(L)-J,2*L:PRINT " ";CHR$(
7);
170 FOR TE=1 TO 200:NEXT TE
180 NEXT J
190 T=T+C
200 F(L)=F(L)+C
210 B=1+B MOD 2
220 IF T<80 THEN 100
230 LOCATE 12,25:PAPER 3:PEN B:PRINT "LE
 JOUEUR";B;"GAGNE":SOUND 1,478:SOUND 1,3
19:SOUND 1,379
240 GOTO 240
```

Musique

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu éducatif.
Difficulté du jeu : moyenne.

LE JEU :
Il se présente comme un exercice d'apprentissage du solfège : l'appareil annonce une note (en clé de sol), suivie de la mention GRAVE ou AIGUE, selon sa place sur la portée. Il faudra répondre en indiquant sur quelle ligne la note sera affichée.
La note est alors dessinée sur la portée, suivie d'un message BRAVO ou PERDU, selon que l'on avait trouvé juste ou non.

LE PROGRAMME :
La boucle I (6Ø-8Ø) dessine la portée. Une deuxième boucle I (9Ø-12Ø) affiche les numéros des lignes.
La ligne 13Ø définit le nom des notes, et la ligne 14Ø réalise le tirage aléatoire d'une note.
Le nom de la note est calculé à la ligne 15Ø et il est affiché par la ligne 16Ø, suivi de la mention GRAVE ou AIGUE, selon le cas.
La réponse sera entrée en ligne 18Ø.
La ligne 19Ø s'occupe du décodage de la réponse.
Les lignes 2ØØ à 21Ø dessinent la note.
La ligne 23Ø regarde si l'on a gagné ou perdu et affiche le message correspondant.
La ligne 24Ø permet de faire un nouvel essai.

POUR JOUER :
On lance le programme par RUN.
On obtient alors l'affichage de la portée, ainsi que le nom de la note à placer et la question NUMERO DE LA LIGNE.
On répond en donnant un numéro de ligne (de 1 à E) suivi de ENTER.
La note est alors dessinée et le message BRAVO ou PERDU apparaît, dans ce cas, à l'écran.
On doit taper une touche quelconque pour continuer, et s'arrêter en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
On peut étendre à une autre clé que la clé de sol, on peut introduire de la couleur, des sons, etc.

```
10 REM MUSIQUE
20 DEFINT X
30 RANDOMIZE (TIME)
40 INK 0,22:INK 1,7:INK 2,24:INK 3,15
50 CLS
60 FOR I=1 TO 5
70 LOCATE 4,2*I+4:PRINT STRING$(31,154)
80 NEXT I
90 FOR I=1 TO 14
100 LOCATE 1,I+2
110 IF I<=9 THEN PRINT I ELSE PRINT " ";
CHR$(I+55)
120 NEXT I
130 N$="SI LA SOLFA MI RE DO "
140 X=1+RND*13
150 T=(X-1) MOD 7
160 LOCATE 1,18:PRINT "NOTE ";MID$(N$,3*
T+1,3);
170 IF X<=7 THEN PRINT " AIGUE" ELSE PRI
NT " GRAVE"
180 PRINT:INPUT "NUMERO DE LA LIGNE :";L
$
190 L=-(ASC(L$)-48)*(L$<":")-(ASC(L$)-55
)*(L$>="A")
200 PEN 3:LOCATE 18,X:PRINT CHR$(22)+CHR
$(1);CHR$(211);CHR$(8);CHR$(205);CHR$(8)
;CHR$(195)
210 LOCATE 18,X+1:PRINT CHR$(211);CHR$(8
);CHR$(193):LOCATE 17,X+2:PRINT CHR$(143
);CHR$(211);CHR$(22)+CHR$(0);:PEN 1
220 LOCATE 18,24
230 PAPER 3:PEN 2:IF L=X THEN PRINT "BRA
VO"; ELSE PRINT "PERDU";
240 PAPER 0:PEN 1:LOCATE 5,25:INPUT "PRE
SSER <ENTER> POUR CONTINUER";R$:GOTO 50
```

Poker

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu contre AMSTRAD.
Difficulté du jeu : facile.

LE JEU :
C'est une simulation d'une partie de poker traditionnelle, effectuée avec un jeu de 32 cartes, dans laquelle le joueur aura, pour partenaire, AMSTRAD.
Le joueur recevra 5 cartes, repérées par des numéros de 1 à 5, et pourra décider de changer certaines cartes en annonçant celles qu'il écarte. Sa nouvelle donne sera alors affichée, et c'est ensuite au tour de l'AMSTRAD de jouer.

LE PROGRAMME :
Les lignes 5Ø et 6Ø définissent les cartes. Une variable N est dimensionnée en ligne 4Ø, pour recevoir 1Ø cartes joueur et 5 cartes AMSTRAD.
Le tirage sera effectué par une boucle I (8Ø-14Ø) et le contrôle par une boucle J (11Ø-13Ø).
L'affichage des cartes est réalisé par le sous-programme 26Ø-37Ø.
Une autre boucle I (19Ø-21Ø) réalise le deuxième tirage du joueur, la ligne 24Ø permettant le deuxième affichage, et la boucle I (26Ø-33Ø) où l'on entre directement après, réalisera l'affichage du jeu machine.

POUR JOUER :
L'ordre RUN lance l'exécution du programme, et les 5 premières cartes du joueur sont alors affichées.
A la question CARTES REFUSEES, le joueur peut répondre Ø suivi de ENTER s'il veut arrêter la partie. Pour jouer, il répondra simplement ENTER s'il s'estime servi, ou il entrera les numéros des cartes qu'il refuse suivis de ENTER.
Par exemple, s'il tape 235 et ENTER, son nouveau jeu conservera les cartes 1 et 4. Son jeu sera alors affiché ainsi que celui de l'AMSTRAD, ce dernier n'ayant pas, dans ce programme, la possibilité de reprendre des cartes.
Il suffira ensuite de taper une touche quelconque pour démarrer une autre partie.

EXTENSIONS POSSIBLES :
Elles seront, bien sûr, très nombreuses, depuis un affichage graphique en couleur des cartes jusqu'à l'amélioration du jeu de l'AMSTRAD, l'introduction des règles sur les paris et même, pourquoi pas, des possibilités de bluff.

```
10 REM POKER
20 RANDOMIZE(TIME)
30 INK 0,22:INK 1,2:INK 2,6:INK 3,0
40 DIM N(32)
50 V$="7 8 9 10VADAROAS"
60 C$=CHR$(226)+CHR$(227)+CHR$(228)+CHR$
(229)
70 CLS
80 FOR I=1 TO 15
90 N(I)=INT(RND*32)
100 IF I=1 THEN 140
110 FOR J=1 TO I-1
120 IF N(I)=N(J) THEN 90
130 NEXT J
140 NEXT I
150 K=0:P=0
160 LOCATE 1,12:PRINT "CARTES REFUSEES"
170 GOSUB 260
180 IF R$="0" THEN END
190 FOR I=1 TO MIN (5,LEN(R$))
200 N(VAL(MID$(R$,I,1)))=N(I+5)
210 NEXT I
220 CLS
230 P=1
240 GOSUB 260
250 K=K+10
260 FOR I=1 TO 5
270 C=INT(N(I+K)/8)
280 F=N(I+K) MOD 8
290 R$=MID$(V$,2*F+1,2)
300 LOCATE 7*I-2,K+1:PRINT I
310 LOCATE 7*I-3,K+3:PRINT R$;": ";:PEN
(3+(C=1 OR C=2)):PRINT MID$(C$,C+1,1);:P
EN 1
320 LOCATE 7*I-3,K+4:PRINT STRING$(5,208
);
330 NEXT
340 IF P=1 THEN P=0:RETURN
350 IF K=10 THEN LOCATE 6,25:INPUT "FAIR
E <ENTER> POUR CONTINUER",R$:GOTO 70
360 LOCATE 18,12:INPUT R$
370 RETURN
```

Le jeu de saute-mouton

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : facile.

LE JEU :
Bien que l'idée en soit très simple, il ne sera pas forcément évident de découvrir une stratégie convenable. On dispose de 3 moutons bleus et de 3 moutons rouges, représentés par des symboles graphiques, qui occupent 8 cases ; les moutons peuvent se déplacer d'une case à une case libre. Plusieurs règles pourront intervenir, selon votre propre estimation de l'intérêt de telle ou telle : soit l'on contraint les moutons à se placer sur la première case disponible, en sautant éventuellement par-dessus les moutons adverses, soit on leur laisse choisir leur case d'arrivée, ce qui peut rendre les parties interminables, à vous de choisir.

LE PROGRAMME :
La boucle I (7Ø-1ØØ) initialise une table A (I), qui contient les positions des moutons. La boucle I (12Ø-17Ø) dessine le terrain de jeu et les moutons.
Les lignes 19Ø et 21Ø s'occupent du texte et des entrées.
La fenêtre # 1, définie en ligne 5Ø, sert à préserver le dessin du terrain de jeu pendant la saisie en ligne 21Ø.
Les lignes 22Ø et 23Ø contrôlent la validité du coup. Le changement de joueur est réalisé en ligne 26Ø.

POUR JOUER :
On commence l'exécution du programme par RUN. On obtient alors un affichage JOUEUR 1, qui indique le numéro du joueur, suivi de la question DEBUT, FIN.
On répond en indiquant le numéro du pion à déplacer suivi de ENTER, puis le numéro de la case d'arrivée, suivi de ENTER.
Le déplacement est alors réalisé ; puis le deuxième joueur procède de la même façon. Le gagnant est en principe celui qui réussit à placer ses trois moutons côte à côte sur les emplacements initialement occupés par l'adversaire.

EXTENSIONS POSSIBLES :
On commencera par rajouter un contrôle du joueur gagnant.
On peut également améliorer la présentation en dessinant de vrais moutons, rendre le jeu plus attractif en les numérotant, et en imposant un certain ordre à l'arrivée.
Enfin, on pourra définir une stratégie pour l'ordinateur, en figeant les règles.

```
10 REM LE JEU DE SAUTE-MOUTON
20 CLS
30 INK 0,9:INK 3,3
40 DIM A(8)
50 WINDOW #1,1,40,13,13
60 A(1)=0:A(8)=0
70 FOR I=1 TO 3
80 A(I+1)=1
90 A(I+4)=-1
100 NEXT I
110 H=1
120 FOR I=1 TO 8
130 LOCATE 4*I,8:PEN 1:PRINT I
140 LOCATE 4*I+1,6:PRINT " "
150 IF A(I)=1 THEN LOCATE 4*I+1,6:PEN 2:
PRINT CHR$(189)
160 IF A(I)=-1 THEN LOCATE 4*I+1,6:PEN 3
:PRINT CHR$(191)
170 NEXT I
180 J=1-(H=-1)
190 LOCATE 1,11:PEN J+1:PRINT "JOUEUR";J
RUN
200 WINDOW SWAP 0,1:PEN J+1
210 INPUT "DEBUT,FIN:";D,F
220 IF D<1 OR D>8 OR F<1 OR F>8 THEN 210
230 IF A(D)<>H OR A(F)<>0 THEN 210
240 A(D)=0
250 A(F)=H
260 H=-H
270 WINDOW SWAP 1,0:GOTO 120
```

Lettres

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
La règle est fortement inspirée d'un jeu de lettres très connu en France, pratiqué notamment dans une émission télévisée (les chiffres et les lettres). AMSTRAD choisit au hasard un certain nombre de symboles (voyelles et consonnes) dans l'alphabet latin, et chacun des joueurs devra indiquer la longueur du mot qu'il a réussi à obtenir (l'appareil ne vérifiera pas si le mot est correct !).
Le score total de chaque joueur est affiché ensuite.

LE PROGRAMME :
La boucle I (90-120) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 60. Pour éviter d'avoir trop de consonnes, on utilise en ligne 110 une petite astuce, qui consiste à refaire au hasard un tirage dans la table A$, contenant des voyelles (on peut en avoir plus ou moins en utilisant d'autres RND).
La boucle 130-170 demande à chacun des joueurs la longueur de son mot.
Les lignes 180 et 190 comparent les résultats et totalisent le score de chaque joueur, affiché par les lignes 200 et 210.

POUR JOUER :
On lance l'exécution du jeu avec RUN.
Les lettres sont alors affichées, puis la question JOUEUR 1 LONGUEUR à laquelle le premier joueur répond en donnant la longueur du mot qu'il a obtenu (l'autre joueur devra toutefois vérifier si le mot est correct) ; puis le joueur 2 fera de même et le score de chaque joueur sera alors affiché.
Pour continuer la partie, il suffira d'appuyer sur une touche quelconque.
On devra faire ESC deux fois pour finir.

EXTENSIONS POSSIBLES :
On pourra bien entendu songer à introduire une temporisation avant de demander leurs réponses aux joueurs.
On pourra aussi entrer non point la longueur du mot, mais le mot lui-même, l'adversaire devant alors répondre à la question CORRECT OU NON ? On peut aussi étendre le jeu à plus de deux joueurs. Et pourquoi ne pas essayer de définir un jeu contre AMSTRAD, auquel on aurait auparavant entré une liste de mots assez longue, dont il afficherait les lettres dans le désordre, avec éventuellement quelques lettres de complément choisies au hasard ?

```
10 REM LETTRES
20 RANDOMIZE(TIME)
30 DEFINT L
40 S=0:J=0
50 CLS
60 L=RND*5+7
70 A$="AEIOUEA"
80 LOCATE 1,6:PRINT "LETTRES :"
90 FOR I=1 TO L
100 LOCATE 8+2*I,6:PRINT CHR$(INT(RND*26
+65))
110 IF RND<0.4 THEN LOCATE 8+2*I,6:PRINT
 MID$(A$,INT(RND*7+1),1)
120 NEXT I
130 I=1
140 LOCATE 1,6+4*I:PEN I+1:PRINT "JOUEUR
";I;:INPUT "- LONGUEUR :";L:PEN 1
150 IF I=1 THEN P=L
160 I=I+1
170 IF I<3 THEN 140
180 S=S+(P AND P>L)
190 J=J+(L AND L>P)
200 LOCATE 1,18:PRINT "SCORE - JOUEUR 1:
";:PEN 2:PRINT S;:PEN 1
210 PRINT "JOUEUR 2:";:PEN 3:PRINT J:PEN
 1
220 LOCATE 5,25:PAPER 3:INPUT "TAPER <EN
TER> POUR CONTINUER";R$:PAPER 0
230 GOTO 50
```

Machinamot

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de hasard (sur des mots).
Difficulté du jeu : aucune.

LE JEU :
Ce très court programme permet à votre AMSTRAD de faire la démonstration de ses capacités à manipuler des chaînes de caractères : on part simplement d'un préfixe et d'un suffixe d'origine grecque, que l'on va assembler au hasard, on tombera bien sûr, de temps à autre, sur des mots absurdes où même légèrement équivoques.
Un nouveau mot est confectionné à chaque essai, mais il est possible d'obtenir plusieurs fois le même mot.

LE PROGRAMME :
Deux variables A$ et B$ sont dimensionnées à 14 en ligne 6Ø et 7Ø : l'une contiendra les préfixes, l'autre les suffixes (si l'on désire rajouter des possibilités, il suffira de modifier cette limitation à 14).
La boucle I (8Ø-1ØØ) lit les lignes DATA (18Ø-2ØØ) concernant les préfixes suivis des suffixes.
La ligne 11Ø effectue le choix d'un préfixe au hasard.
La ligne 12Ø effectue le choix d'un suffixe au hasard.
Le mot obtenu en ligne 13Ø est affiché par la ligne 14Ø, puis on recommence (ligne 15Ø).
La boucle WHILE/WEND en 16Ø provoque une attente qui utilise l'horloge interne (TIME) d'AMSTRAD.

POUR JOUER :
Lancer l'exécution du programme par RUN. Un premier mot sera immédiatement affiché, puis un nouveau mot sera affiché à la suite du précédent.
Taper ESC deux fois pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra très facilement augmenter la liste des mots disponibles.
On pourra aussi utiliser un procédé analogue pour faire confectionner des phrases à l'ordinateur, lui faire inventer des poèmes, etc. On pourra aussi transformer le programme initial en un jeu de société qui consisterait, par exemple, à prendre des paris sur le prochain mot qui va apparaître. Et confier, éventuellement, la gestion des paris à votre AMSTRAD...

```
10 REM MACHINAMOT
20 RANDOMIZE(TIME)
30 DEFINT I,J,P
40 MODE 0
50 CLS
60 DIM A$(14)
70 DIM B$(14)
80 FOR I=0 TO 14
90 READ A$(I),B$(I)
100 NEXT I
110 I=RND*14
120 J=RND*14
130 C$=A$(I)+B$(J)
140 P=1+RND*12:PEN P:PRINT C$
150 X=TIME
160 WHILE TIME-X<100:WEND
170 GOTO 110
180 DATA ANDRO,CRATE,COSMO,PHILE,MICRO,P
HOBE,HELIO,TROPE,SEXO,GYNE,STEREO,MANE
190 DATA PALEO,NAUTE,MACRO,PHAGE,HYDRO,F
UGE,STENO,PATHE,PATHO,CARDE,OLEO,DERME
200 DATA MYTHO,SCOPE,HOMEO,VORE,OCTO,PHO
NE
```

Espion

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : facile.

LE JEU :
Le programme peut servir simultanément de codeur et de décodeur de message secret, suivant une méthode très facile à mettre en œuvre sur ce type d'appareil.
Un petit générateur aléatoire situé en fin de programme est utilisé pour modifier de façon arbitraire chaque lettre du message ; on devra bien veiller à utiliser toujours le même nombre (noyau) pour commencer le codage ou le décodage de deux messages identiques.

LE PROGRAMME :
Le noyau initial (qui sera un entier quelconque) est entré en ligne 3Ø ; le message est entré à la ligne 5Ø.
La ligne 6Ø permet de choisir entre les options de codage et de décodage.
La boucle I (8Ø-15Ø) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.
La ligne 1ØØ appelle le générateur aléatoire.
La ligne 11Ø réalise la transposition des lettres.
Les lignes 12Ø et 13Ø vérifient que l'on reste bien dans les limites autorisées.
Les résultats seront imprimés par la ligne 16Ø.
Les lignes 1ØØØ et 1Ø3Ø contiennent le générateur.

POUR JOUER :
Lancer le programme par RUN.
Entrer ensuite un entier N qui servira de noyau pour le générateur (il faudra le retenir pour le décodage).
Puis taper le message, suivi de ENTER.
Choisir l'option C pour CODER, et l'option D pour DECODER, et taper ENTER.
Le message codé ou décodé, selon le cas, sera alors affiché.

EXTENSIONS POSSIBLES :
On pourra étendre le codage à d'autres signes que les lettres de l'alphabet, utiliser éventuellement plusieurs générateurs, introduire dans le message un codage du ou des nombres ayant servi de générateurs, etc.

```
10 REM ESPION
20 CLS
30 INPUT"DONNEZ VOTRE CODE :";N
40 A=1/N
50 PRINT:INPUT "VOTRE MESSAGE :";A$
60 PRINT:PRINT "VOULEZ-VOUS ";:PEN 2:PRI
NT "(C)ODER";:PEN 1:PRINT " OU ";:PEN 3:
PRINT "(D)ECODER";:PEN 1:INPUT " :";R$
70 B$=""
80 FOR I=1 TO LEN(A$)
90 T=ASC(MID$(A$,I,1))
100 GOSUB 1000
110 T=T-A*((R$="C")-(R$="D"))
120 IF T>90 THEN T=T-26
130 IF T<65 THEN T=T+26
140 B$=B$+CHR$(T)
150 NEXT I
160 PRINT:PRINT "RESULTAT :";:PEN (2-(R$
="D")):PRINT B$:PEN 1:END
1000 A=(A+PI)↑8
1010 A=A-INT(A)
1020 A=INT(26*A+1)
1030 RETURN
```

Jeu de loto

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

LE JEU :
Il s'agit de réaliser un tirage complet de loto, c'est-à-dire d'obtenir 6 numéros, compris entre 1 et 49, plus un numéro complémentaire, tous différents, qui pourraient être, par exemple, utilisés pour remplir votre prochaine grille de loto.

LE PROGRAMME :
On utilisera un tableau N(7), dimensionné en ligne 4Ø, pour enregistrer les résultats.
Le tirage des nombres est effectué dans une boucle I (6Ø-12Ø), une deuxième boucle J (9Ø-11Ø) étant utilisée pour contrôler qu'il n'y a pas deux tirages identiques.
Le tirage (ou le retirage) est effectué en ligne 7Ø.
La boucle I (15Ø-18Ø) affiche le résultat, et la ligne 19Ø affiche le numéro complémentaire.
Les lignes 21Ø à 23Ø s'occupent du cas où l'on voudrait refaire une autre partie.

POUR JOUER :
Il suffit de taper RUN pour lancer le programme et obtenir l'affichage d'un premier loto.
La question AUTRE TIRAGE (O/N) s'affiche en bas de l'écran ; on répondra O si l'on désire continuer.

EXTENSIONS POSSIBLES :
On peut imaginer un programme qui pourrait tenir compte, soit de ses propres résultats, soit de résultats entrés par un joueur, pour améliorer le tirage, en tenant compte des probabilités de sortir un numéro donné.
On peut aussi envisager un programme qui réaliserait des cartons pour le jeu de loto plus usuel, tel qu'il est encore couramment pratiqué dans le Midi de la France.

```
10 REM JEU DE LOTO
20 RANDOMIZE(TIME)
30 MODE 0
40 DIM N(7)
50 CLS
60 FOR I=1 TO 7
70 N(I)=1+INT(RND*49)
80 IF I=1 THEN 120
90 FOR J=1 TO I-1
100 IF N(I)=N(J) THEN 70
110 NEXT J
120 NEXT I
130 LOCATE 4,6:PEN 1:PRINT "TIRAGE DU LO
TO"
140 PRINT STRING$(3," "); STRING$(14,"-"
)
150 FOR I=1 TO 6
160 LOCATE 3*I-2,12:PEN 3+I
170 PRINT N(I)
180 NEXT I
190 LOCATE 1,18:PEN 1:PRINT "COMPLEMENTA
IRE :";:PEN 15:PRINT N(7):PEN 1
200 SOUND 1,478:SOUND 1,426:SOUND 1,379
210 LOCATE 1,24:PAPER 1:PEN 0:PRINT "aut
re tirage (O/N)?":PAPER 0:PEN 1
220 A$=INKEY$:IF A$="" THEN 220
230 IF UPPER$(A$)="O" THEN 50
```

Tirage d'un dé truqué

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Il reprend une idée d'un programme précédent (DE TRUQUE) dans une application statistique : on fabrique un dé volontairement truqué, que l'on va lancer un certain nombre de fois, et l'on dresse un tableau des résultats.

LE PROGRAMME :
Les probabilités désirées pour chaque tirage sont introduites par une boucle I (7Ø-14Ø)avec toutes les vérifications de validité nécessaires. Le nombre de tirages demandés est introduit en ligne 16Ø.
La boucle J (17Ø-25Ø) comptabilise les tirages.
La boucle I (19Ø-21Ø) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.
Les résultats sont comptabilisés en ligne 22Ø, et affichés grâce aux lignes 23Ø et 24Ø.

POUR JOUER :
Commencer l'exécution en frappant RUN, puis à la question PROBABILITE DU DE 1, introduire un nombre compris entre Ø et 1.
Faire de même pour les dés 2, 3, etc., sans oublier que le total des probabilités introduites devra être proche de 1.
Ensuite, à l'affichage de la question NOMBRE DE TIRAGES, on introduira un entier suffisamment grand, suivi de ENTER. L'affichage des résultats mettra d'autant plus de temps à apparaître que l'entier introduit sera grand.
Les résultats sont affichés sous les numéros des tirages : ainsi 32.3 apparaissant sous le 1 signifie que le tirage de 1, auquel on avait affecté une probabilité de Ø.3, a été réalisé 32 fois.

EXTENSIONS POSSIBLES :
On peut envisager d'améliorer la représentation des résultats en utilisant par exemple un histogramme, ou toute autre méthode graphiquement plus parlante.

```
10 REM TIRAGE D'UN DE TRUQUE
20 RANDOMIZE(TIME)
30 INK 0,2:INK 1,17
40 DIM E(6)
50 CLS
60 P=0
70 FOR I=1 TO 6
80 LOCATE 1,3:PRINT STRING$(40,32)
90 LOCATE 1,3:PRINT "PROBABILITE DU DE";
I;:INPUT E(I)
100 P=P+E(I)
110 IF E(I)<0 OR E(I)>=1 OR P>1.1 THEN 3
00
120 LOCATE 6*I-3,6:PRINT "DE";I
130 LOCATE 6*I-3,8:PRINT E(I)
140 NEXT I
150 IF P<0.9 THEN RUN
160 PRINT:PRINT:INPUT "NOMBRE DE TIRAGES
 :";P
170 FOR J=1 TO P
180 N=INT(RND*6+1)
190 FOR I=1 TO 6
200 IF N=I AND RND>E(I)-INT(E(I)) THEN 1
80
210 NEXT I
220 E(N)=E(N)+1
230 LOCATE 6*N-1,15:PRINT N
240 LOCATE 6*N-3,17:PRINT E(N)
250 NEXT J
260 END
300 PAPER 3:LOCATE 11,25:PRINT CHR$(7);"
IMPOSSIBLE :0<=P<1":PAPER 0
310 P=P-E(I)
320 FOR J=1 TO 1000:NEXT J
330 LOCATE 10,25:PRINT STRING$(22,32);
340 GOTO 80
```

Le jeu de Marienbad

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : logique (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Un jeu rendu très célèbre grâce à un film qui le popularisa, présenté toutefois ici dans une version plus conforme à ses origines, à savoir le jeu de NIM.
En effet, il est possible, en début de partie, de choisir le nombre d'allumettes que l'on veut mettre dans chaque tas.
Chaque joueur joue à son tour, et celui qui retire la dernière allumette a perdu.

LE PROGRAMME :
La ligne 6Ø dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.
La boucle I (7Ø-1ØØ) permet d'entrer ce nombre.
La boucle I (13Ø-16Ø) imprime les tas.
La ligne 17Ø assure l'affichage du joueur.
La ligne 19Ø, ainsi que la ligne 2ØØ, contrôlent si le coup est autorisé.
La ligne 26Ø contrôle la fin de la partie.

POUR JOUER :
On lance le programme avec RUN, puis on entre le nombre de tas (entre 1 et 7), suivi de ENTER.
On entrera ensuite le nombre d'allumettes dans chaque tas (faire à chaque entrée N suivi de ENTER). On obtiendra alors l'affichage des différents tas, avec le numéro du joueur concerné (1 ou 2), ainsi que le message DANS QUEL TAS ; on répond en tapant le numéro du tas dans lequel on désire faire un retrait suivi de ENTER. On obtient ensuite la question QUANTITE A ENLEVER, et l'on répond en indiquant le nombre d'allumettes que l'on désire enlever du tas concerné.
Puis ce sera au tour du deuxième joueur, et l'on jouera alternativement jusqu'à ce qu'il n'y ait plus d'allumettes disponibles.
En fin de partie, on obtiendra l'affichage du numéro du joueur gagnant.

EXTENSIONS POSSIBLES :
On peut désigner chaque joueur par son nom, faire un graphisme plus élaboré, écrire un programme permettant à votre AMSTRAD de servir de partenaire, etc.

```
10 REM LE JEU DE MARIENBAD
20 CLS
30 PRINT "NOMBRE DE TAS ( <=7 ):";:PAPER
 3:INPUT "",N:PAPER 0
40 IF N>=8 OR N<=0 THEN RUN
50 S=0
60 DIM A(N)
70 FOR I=1 TO N
80 PRINT:PRINT"TAS ";I;:INPUT "- NOMBRE
D'ALLUMETTES:";A(I)
90 S=S+A(I)
100 NEXT I
110 H=1
120 CLS
130 FOR I=1 TO N
140 LOCATE 5*I+3,5:PAPER 1:PEN 0:PRINT U
SING "####";A(I)
150 PAPER 0:PEN 1:PRINT:PRINT "TAS":LOCA
TE 5*I+3,7:PRINT I
160 NEXT I
170 LOCATE 1,10:PEN 1+H:PRINT "JOUEUR";H
180 PRINT:INPUT "DANS QUEL TAS       :";T
190 IF T>N THEN PRINT CHR$(7);:GOTO 120
200 IF A(T)=0 THEN PRINT CHR$(7);:GOTO 1
20
210 LOCATE 1,14:PRINT CHR$(20);:INPUT "Q
UANTITE A ENLEVER :";Q
220 IF Q>A(T) OR Q<=0 THEN PRINT CHR$(7)
;:GOTO 200
230 A(T)=A(T)-Q
240 S=S-Q
250 H=1+H MOD 2
260 IF S=0 THEN PRINT:PRINT "LE JOUEUR "
;H;"A GAGNE.":SOUND 1,478:SOUND 1,319:SO
UND 1,379:END
270 GOTO 120
```

Compose

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : composition musicale.
Difficulté du jeu : aucune, a priori.

LE JEU :
Il s'agit, en fait, d'un programme capable de composer, de façon très élémentaire, de la musique, et ceci d'une façon complètement aléatoire. L'ordinateur va, dans un premier temps, dessiner une portée, puis il affiche une succession de notes (allant du DO grave au SI aigu) et donne leur nom. Une nouvelle série de notes sera ensuite affichée, et ainsi de suite.

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine la portée. La variable N$, en ligne 8Ø, contient les noms des notes et V$ contient les notes sous forme codée.
La deuxième boucle I (9Ø-15Ø) permet de faire le choix d'une note de façon aléatoire, à la ligne 1ØØ ; son nom est calculé en ligne 11Ø et la note est dessinée par la ligne 12Ø.
La ligne 13Ø joue la note, tandis que la boucle WHILE/WEND interdit de continuer jusqu'à la fin de son exécution.
La ligne 16Ø permet de recommencer en tapant ENTER.

POUR JOUER :
Lancer l'exécution du programme en tapant RUN.
La portée sera alors rapidement affichée, puis les notes sont représentées, avec leur nom affiché en bas de l'écran.
Une pression sur une touche quelconque provoquera l'affichage d'une nouvelle série de notes, et une double pression sur la touche ESC provoque la fin de la partie.

EXTENSIONS POSSIBLES :
On pourra affiner les règles de composition, afficher davantage de notes, introduire des dièses et des bémols, améliorer le graphisme.

```
10 REM COMPOSE
20 RANDOMIZE(TIME)
30 PRINT CHR$(22);CHR$(1)
40 CLS
50 FOR I=1 TO 5
60 LOCATE 1,4+2*I:PEN 3:PRINT STRING$(40
,154)
70 NEXT I
80 N$="SI LA SOLFA MI RE DO ":V$="127142
159179190213239253284319358379426478"
90 FOR I=1 TO 10
100 X=INT(RND*14+1)
110 D$=MID$(N$,3*(X+7*(X>7))-2,3)
120 PEN 2:LOCATE 4*I-2,X:PRINT CHR$(211)
;CHR$(10);CHR$(8);CHR$(211);CHR$(10);CHR
$(8);CHR$(8);"O"
130 SOUND 1,VAL(MID$(V$,3*X-2,3)):WHILE
SQ(1)>127:WEND
140 PEN 1:LOCATE 4*I-3,20:PRINT D$
150 NEXT I
160 PRINT CHR$(22)+CHR$(0);:PAPER 1:PEN
0:LOCATE 7,24:INPUT "TAPER <ENTER> POUR
CONTINUER",R$:PAPER 0
170 GOTO 40
```

Jour de la semaine

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : utilitaire.
Difficulté du jeu : facile.

LE JEU :
Le programme vous permet de retrouver, sans aucun calcul, le jour de la semaine correspondant à une date donnée.
La date sera entrée sous forme de trois nombres, par exemple 11, 3, 1956 pour le 11 mars 1956, et le résultat sera affiché en clair.

LE PROGRAMME :
Les lignes 2Ø à 6Ø contrôlent les entrées. La ligne 4Ø s'occupe de la fin de partie. La ligne 9Ø permet une correction pour les dates antérieures à 1582 (mise en service du calendrier actuel).
Les lignes 1ØØ à 11Ø permettront l'affichage en clair du mois dans la réponse (sur 4 lettres).
La formule utilisée apparaît aux lignes 13Ø et 14Ø.
Les lignes 15Ø à 18Ø permettront l'affichage en clair du jour dans la réponse. La ligne 18Ø imprime les résultats centrés à l'écran.

POUR JOUER :
Lancer le programme par RUN, puis entrer le jour suivi de ENTER, le mois (entre 1 et 12) suivi de ENTER, et l'année (sous la forme 1845, par exemple) suivie de ENTER.
Le résultat est très rapidement affiché.
On tape une touche quelconque et le programme revient sur la question JOUR.
Pour continuer, il suffit de retaper un jour, un mois, une année.
Pour terminer, il faut répondre Ø à la question JOUR.

EXTENSIONS POSSIBLES :
On pourra entrer les mois en clair et faire afficher le nom du jour en entier.
On peut envisager de faire afficher les dates et fêtes particulières (Noël, Pâques, les Rameaux, Pentecôte...).
On peut écrire un programme qui afficherait les jours fériés pour une année donnée, etc.

```
10 REM JOUR DE LA SEMAINE
20 CLS:INK 3,6,3
30 PRINT:INPUT "JOUR . . . . . . . . . . :";
J
40 IF J=0 THEN END
50 PRINT:INPUT "MOIS . . . . . . . . . . :";
M
60 PRINT:INPUT "ANNEE. . . . . . . . . . :";
A
70 B=A
80 S=INT(A/100)-INT(A/400)
90 IF A<=1582 THEN S=2
100 M$="JANVFEVRMARSAVRIMAI JUINJUILAOUT
SEPTOCTONOVEDECE"
110 M$=MID$(M$,4*M-3,4)
120 IF M<3 THEN GOSUB 220
130 S=J-1+INT(13*(M+1)/5)+INT(5*A/4)-S
140 S=S MOD 7
150 J$="DIMANCHELUNDI   MARDI   MERCREDI
JEUDI   VENDREDISAMEDI  "
160 J$=MID$(J$,8*S+1,8):WHILE RIGHT$(J$,
1)=" ":J$=LEFT$(J$,LEN(J$)-1):WEND
170 P$="LE"+STR$(J)+" "+M$+STR$(B)+" ETA
IT UN "
180 PEN 2:LOCATE (40-LEN(P$)-LEN(J$))/2,
20:PRINT P$;:PEN 3:PRINT J$:PAPER 1:PEN
0
190 SOUND 1,478:SOUND 1,319:SOUND 1,379
200 LOCATE 6,24:INPUT "TAPER <ENTER> POU
R CONTINUER",R$:PAPER 0:PEN 1
210 RUN
220 M=M+12
230 A=A-1
240 RETURN
```

Tour de cartes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de prestidigitation.
Difficulté du jeu : facile.

LE JEU :
C'est un tour assez facile mais, ici, les cartes sont manipulées sous vos yeux par AMSTRAD qui tient le rôle du prestidigitateur ; 21 cartes, prises au hasard dans un paquet de 52, sont affichées, et le joueur devra retenir mentalement l'une des cartes présentées. Puis, il devra indiquer à votre AMSTRAD dans quelle colonne (1re, 2e ou 3e à partir de la gauche) se trouve la carte choisie ; les cartes sont alors redistribuées et le joueur indique de nouveau la colonne correspondant à sa carte ; ceci est fait encore une fois et, au 4e affichage, AMSTRAD est en mesure de deviner la carte choisie.

LE PROGRAMME :
La boucle I (60-100) permet de choisir, au hasard, 21 cartes sur 52, et la ligne 80 permet de s'assurer que l'on ne réalise pas deux fois le même tirage. La ligne 120 contient les noms des cartes (2, 3... RO, AS) et leur couleur dans deux variables alphanumériques V$ et C$.
La boucle I (130-210) réalise l'affichage du jeu.
La ligne 140 décode la couleur de la carte.
La ligne 150 choisit la position d'affichage.
La ligne 160 décode la hauteur.
L'affichage proprement dit est effectué de la ligne 170 à la ligne 200.
La ligne 230 permet au joueur d'entrer le numéro de la colonne où se trouve sa carte. La boucle J (270-290) réalise le brouillage, et les boucles emboîtées J (320-360) et I (330-350) fabriquent la nouvelle variable A$ à utiliser pour le prochain affichage.

POUR JOUER :
On lance le programme par RUN.
On obtient un premier affichage de 21 cartes réparties sur 3 colonnes.
On indique alors le numéro (1, 2 ou 3) de la colonne où se trouve la carte choisie, suivi de ENTER.
On obtient un 2e affichage et l'on fait de même ; puis un 3e affichage et l'on fait de même.
Au 4e affichage, la carte choisie est automatiquement affichée.
On sortira du programme en tapant ESC deux fois à la place du numéro de la colonne.

EXTENSIONS POSSIBLES :

On peut envisager une amélioration de l'affichage en écrivant les noms des cartes en entier, ou même, en dessinant de véritables cartes, en couleur. On pourra aussi essayer de programmer d'autres tours, utilisant le même principe de base, mais avec davantage de cartes, etc.

```
10 REM TOUR DE CARTES
20 RANDOMIZE (TIME)
30 BORDER 9:INK 0,9:INK 1, 26:INK 2,0:IN
K 3,6
40 CLS
50 A$=""
60 FOR I=1 TO 21
70 B$=CHR$(INT(RND*52+38))
80 IF INSTR (A$,B$)<>0 THEN 70
90 A$=A$+B$
100 NEXT I
110 H=3:P=1
120 V$="2.3 4 5 6 7 8 9 10VADAROAS":C$=C
HR$(226)+CHR$(227)+CHR$(228)+CHR$(229)
130 FOR I=1 TO LEN(A$)*P-11*(P=0)
140 S=1+INT((ASC(MID$(A$,I,1))-38)/13)
150 J=INT((I-1)/7)
160 T=(ASC(MID$(A$,I,1))-38) MOD 13
170 IF P=0 THEN LOCATE 1,12:PRINT "VOTRE
 CARTE EST :"
180 LOCATE 1+17*J*P-17*(P=0),1+(I-7*J)*P
-11*(P=0)
190 PRINT MID$(V$,2*T+1,2);" ";
200 PEN (2-(S=2 OR S=3)):PRINT MID$(C$,S
,1):PEN 1
210 NEXT I
220 IF P=0 THEN END
230 LOCATE 1,16:INPUT "QUELLE COLONNE :"
;N
240 CLS
250 H=H-1
260 R$="20120":B$=""
270 FOR J=0 TO 2
280 B$=B$+MID$(A$,7*VAL(MID$(R$,N+J,1))+
1,7)
290 NEXT J
300 IF H=0 THEN A$=B$:P=0:GOTO 120
310 A$=""
320 FOR J=0 TO 2
330 FOR I=1 TO 7
340 A$=A$+MID$(B$,3*I-2+J,1)
350 NEXT I
360 NEXT J
370 GOTO 120
```

L'indésirable

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard et réflexion (1 joueur).
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes le propriétaire d'un immeuble de 16 pièces vides, et vous n'arrivez pas à mettre la main sur votre locataire indésirable. Vous visitez les pièces au hasard, mais, si vous visitez la pièce voisine, votre locataire, prévenu par le bruit, va déménager. Il peut même se réinstaller dans une pièce que vous avez déjà visitée, et marquée comme vide. Il vous faudra arriver à le déloger en moins de 9 coups pour gagner.

LE PROGRAMME :
Une table B est dimensionnée à 16, en ligne 5Ø, pour recevoir les numéros des cases déjà visitées.
La boucle I (7Ø-1ØØ) dessine le terrain de jeu à l'aide d'une encre transparente.
La position initiale du locataire est tirée au hasard en ligne 11Ø.
Le numéro de la case à visiter est donné en ligne 12Ø et est vérifié en ligne 13Ø.
Les lignes 14Ø à 21Ø contrôlent les diverses réponses possibles.
La boucle I (18Ø-2ØØ) dessine les cases visitées.
La ligne 22Ø dessine le locataire, effacé en ligne 25Ø.
Les lignes 26Ø et 27Ø assurent son déplacement.

POUR JOUER :
On lance l'exécution par RUN.
A la question CASE VISITEE, on répond en tapant une lettre, de A à P. Si on a trouvé le locataire, celui-ci est affiché, et la partie est terminée ; sinon, deux cas peuvent se produire : soit l'on est suffisamment loin, et le locataire ne bouge pas, soit l'on est à une case, il apparaît un court instant, et il se recache, aléatoirement, dans une case voisine. Les cases déjà visitées se colorent en bleu, et vous ne pouvez pas y revenir. Si, au bout de 9 coups, vous n'avez pas trouvé, la position du locataire s'affiche, et vous avez perdu.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, jouer sur plus de 16 cases, utiliser des caractères graphiques pour représenter le locataire et le propriétaire, utiliser un algorithme plus subtil pour le déplacement du locataire, lui interdire éventuellement de revenir dans des cases déjà visitées, etc.

```
10 REM L'INDESIRABLE
20 RANDOMIZE(TIME)
30 PRINT CHR$(22)+CHR$(1)
40 B$=STRING$(3,127)
50 DIM B(16)
60 CLS
70 FOR I=1 TO 16
80 LOCATE 4+4*(I-1-4*INT((I-1)/4)),2+4*I
NT((I-1)/4):PEN 3:PRINT B$
90 LOCATE 4+4*(I-1-4*INT((I-1)/4)),3+4*I
NT((I-1)/4):PRINT B$;CHR$(8);CHR$(8);:PE
N 2:PRINT CHR$(64+I)
100 NEXT I
110 X=INT(RND*16)+1:S=1:K=0:PRINT CHR$(2
2)+CHR$(0)
120 PEN 1: LOCATE 1,20:PRINT CHR$(20);:I
NPUT "CASE VISITEE :";C$:C=ASC(C$+" ")-6
4
130 IF C<1 OR C>16 THEN 120
140 IF B(C)=1 THEN LOCATE 14,22:PAPER 1:
PEN 0:PRINT "DEJA VISITEE":PRINT CHR$(7)
;:PAPER 0:GOTO 120
150 IF C=X THEN LOCATE 3,22:PEN 2:PRINT
"VOUS L'AVEZ EU EN";S;"COUPS. BRAVO!":SO
UND 1,478:SOUND 1,379:SOUND 1,319:K=1:GO
TO 220
160 IF X-5=C OR X-4=C OR X-3=C OR X-1=C
OR X+1=C OR X+3=C OR X+4=C OR X+5=C THEN
 GOTO 220
170 LOCATE 12,22:PRINT "IL N'EST PAS LA"
:SOUND 1,478:SOUND 1,426:SOUND 1,379
180 S=S+1:B(C)=1:FOR I=1 TO 16
190 LOCATE 4+4*(I-1-4*INT((I-1)/4)),2+4*
INT((I-1)/4):PEN 3-B(I):PRINT B$
200 NEXT I
210 IF S=9 THEN LOCATE 5,22:PAPER 3:PEN
0:PRINT "VOUS AVEZ PERDU, IL ETAIT EN ";
CHR$(X+64):K=1:PAPER 0:PEN 1:ELSE GOTO 1
20
220 LOCATE 5+4*(X-1-4*INT((X-1)/4)),1+4*
INT((X-1)/4):PRINT "%"
230 HE=TIME:WHILE TIME-HE<20:WEND
240 IF K=1 THEN SOUND 1,319:SOUND 1,379:
SOUND 1,478:LOCATE 1,24:END
250 LOCATE 5+4*(X-1-4*INT((X-1)/4)),1+4*
INT((X-1)/4):PRINT " "
260 Z=INT(10*RND+1)-5: IF ABS(Z)=2 THEN
260
270 X=X+Z:IF X>16 OR X<1 THEN X=X-Z:GOTO
 260
280 GOTO 180
```

Le jeu de la vie

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu solitaire (2 versions).
Difficulté du jeu : facile.

LE JEU :
La version présentée ici respecte les règles posées à l'origine par le mathématicien J.H. CONWAY.
Une certaine population est définie au début du jeu soit par AMSTRAD (version automatique) soit par vous-même (version manuelle). Cette population va évoluer sous vos yeux, en respectant certaines règles, très élémentaires de naissance, de vie et de mort. L'évolution se poursuivra jusqu'à ce que l'on décide d'arrêter le programme.

LE PROGRAMME :
Deux tables (B et C) sont utilisées en ligne 5Ø, pour enregistrer l'état de la population. Les boucles F (23Ø-29Ø) et G (24Ø-28Ø) sont utilisées en début de jeu pour définir la population initiale (ligne 25Ø). Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 8Ø-21Ø, 9Ø-2ØØ, 11Ø-16Ø, 12Ø-15Ø. La ligne 13Ø réalise la gestion du nouvel état suivant les règles.
L'affichage de ce nouvel état est obtenu en réutilisant la double boucle d'affichage F (23Ø-29Ø) et G (24Ø-28Ø).
La ligne 3ØØ permet de recommencer.

POUR JOUER :
Dans la version automatique, il suffit de lancer le programme par RUN et de regarder. Pour arrêter, il faudra taper ESC deux fois. Dans la version manuelle, la configuration de départ sera entrée, après le lancement du programme par RUN, en tapant Ø si l'on ne veut rien mettre sur la ligne et la colonne indiquée, et 1 si l'on désire y mettre une cellule. Puis l'évolution se déroulera ensuite de façon automatique.

EXTENSIONS POSSIBLES :
Il sera facile, si l'on dispose d'une mémoire plus importante, d'augmenter les dimensions des tables B et C, en redéfinissant la variable A. On pourra aussi introduire la couleur, améliorer la précision du dessin en remplaçant les PRINT par un affichage haute résolution.

```
10 REM LE JEU DE LA VIE (MANUEL)
30 INK 0,14:INK 1,18
40 INPUT "DIMENSION :";A:IF A<1 OR A>12
THEN RUN
50 DIM B(A,A):DIM C(A,A)
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=
A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 CLS
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 260
255 LOCATE 1,25:PEN 3:PRINT "LIGNE";F;"C
OLONNE";G;"TAPER 1 OU 0        ";
257 LOCATE 33,25:INPUT C(F,G):C(F,G)=-(C
(F,G)=1):PEN 1
260 LOCATE 2*G,2*F:IF C(F,G)=1 THEN PRIN
T "*"
270 B(F,G)=C(F,G)
280 NEXT G
290 NEXT F
300 GOTO 80
```

```
10 REM LE JEU DE LA VIE (AUTOMATIQUE)
20 RANDOMIZE(TIME)
30 INK 0,14:INK 1,18
40 INPUT "DIMENSION :";A:IF A<1 OR A>12
THEN RUN
50 DIM B(A,A):DIM C(A,A)
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=
A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 CLS
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I=0 THEN C(F,G)=-(RND<0.5)
260 LOCATE 2*G,2*F:IF C(F,G)=1 THEN PRIN
T "*"
270 B(F,G)=C(F,G)
280 NEXT G
290 NEXT F
300 GOTO 80
```

Niveau 3

Nom du programme	*Thèmes et particularités*
Losanges	Pour vous aider à vous endormir.
Plus grand, plus petit	Un jeu de déduction intéressant, pour un joueur.
Ordre	Un peu d'astuce pour remettre en ordre, le plus vite possible.
Le pendu	Un jeu de lettres très classique (2 joueurs).
Belote	Pour simuler une donne, avec un jeu de 32 cartes.
Un taquin de calcul	Un bon entraînement (2 joueurs).
Pendu inversé	Cette fois, c'est l'ordinateur qui essaie de deviner le mot.
Nicomaque	Pour se perfectionner en calcul mental (1 ou plusieurs joueurs).
Mastermind	Avec des chiffres, mais très classique. Un seul joueur.
Idem	Un jeu de calcul mental, avec plusieurs niveaux, pour plusieurs joueurs.
Reines sur l'échiquier	Un vieux problème, pour vous aider à réfléchir.
Mots	Une idée très simple, mais ce n'est pas si facile de trouver !
Permutations	Un jeu de réflexion, pas toujours évident.
Bridge	Mettez au point vos propres problèmes de bridge.
Hanoï	Un conseil : commencez petit ! ! !
Course de chevaux	Assez visuel : un jeu de hasard, pour plusieurs joueurs.
Accrochez les wagons	Pour les plus jeunes : très spectaculaire.
Le loup et les agneaux	Un jeu de réflexion pour 2 joueurs.

Manipulation des chaînes et graphiques haute résolution

TRAITEMENT DES CHAINES DE CARACTERES : MID$, LEFT$, RIGHT$

Nous avons eu l'occasion de rencontrer, dès le niveau II, la fonction essentielle utilisée dans le Basic AMSTRAD pour manipuler les chaînes : il s'agit de MID$ (A$, I J) qui permet d'extraire, dans la chaîne A$, une sous-chaîne de longueur J, commençant au Ième caractère.

Ainsi si,

A$ = "MONSIEUR"

on aura :

MID$ (A$, 2, 3) = "ONS"

Pour faciliter les manipulations, il est possible d'utiliser les deux fonctions LEFT$ et RIGHT$.

LEFT$ (A$, I) permet d'extraire les I premiers caractères à partir de la gauche du mot A$.

Ainsi, LEFT$ (A$, 2) = "MO", avec l'exemple précédent (on pourrait d'ailleurs utiliser MID$ (A$, 1, 2) pour obtenir le même résultat).

RIGHT$ (A$, J) permet d'extraire les J derniers caractères (on part de la droite du mot A$).

Ainsi RIGHT$ (A$, 3) = "EUR", toujours avec le même exemple.

Il est plus difficile de se passer de la fonction RIGHT$, mais il serait possible de faire MID$ (A$, LEN (A$) - 3, 3) pour obtenir le même résultat.

Le Basic AMSTRAD permet de plus le remplacement direct d'une sous-chaîne par une affectation du type MID$ (-.-) = "–".

Ainsi, on pourra écrire MID$ (A$, I, 1) = "–", ce qui permet de simplifier considérablement certaines écritures.

LE GRAPHISME AVEC L'INSTRUCTION PLOT

AMSTRAD possède 3 largeurs d'écran (2Ø, 4Ø et 8Ø colonnes) auxquelles correspondent 3 définitions graphiques.

Ces 3 modes ont la particularité de posséder les mêmes repérages d'écran.

Ainsi, en mode Ø, 1 ou 2, l'instruction PLOT X, Y, C imprime un point à la même position d'écran X, Y. Seule varie la dimension du point.

Un paramètre optionnel C permet de varier la couleur d'impression.

LES INSTRUCTIONS DE POSITIONNEMENT : MOVE ET MOVER

L'origine des axes de repérage des coordonnées d'un point graphique est, par défaut, le coin inférieur gauche de l'écran. L'instruction ORIGIN permet de la déplacer.

Dans le repère ainsi défini, MOVE X, Y positionne le curseur graphique, là où les impressions peuvent être faites.

Les déplacements peuvent être effectués relativement à la position actuelle du curseur. Ainsi, MOVER 8, 4, le déplace de 8 unités vers la droite et de 4 vers le haut.

LE DESSIN DE LIGNES : DRAW ET DRAWR

L'instruction DRAW X, Y, dessine une ligne depuis la position du curseur graphique, jusqu'au point de coordonnées absolues X et Y.

L'instruction DRAWR X, Y, dessine le trait jusqu'au point décalé de X unités horizontales et Y unités verticales.

Losanges

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : interlude.
Difficulté du jeu : aucune.

LE JEU :
Le programme va progressivement construire sur l'écran un losange, puis il va l'effacer progressivement, le reconstruire, et ainsi de suite.
Vous pourrez envisager de l'utiliser pour vous endormir, si les moutons ne vous suffisent plus.
Une version, très différente, vous est également proposée, qui vous permettra de mieux comprendre le fonctionnement de la puissante instruction DRAW.

LE PROGRAMME :
La ligne 3Ø met l'origine des axes au centre de l'écran.
Une première boucle K (5Ø-13Ø) définit les dimensions des tracés successifs.
La deuxième boucle I (6Ø-12Ø) trace des bandes de points, en utilisant des instructions PLOT (lignes 8Ø à 11Ø).
Si la variable S (ligne 7Ø) est à −1, alors il y aura effacement aléatoire (on utilise toujours PLOT, mais en se servant de couleurs aléatoires pour effacer).
La boucle WHILE/WEND, initialisée en 14Ø, permet de choisir 4 couleurs différentes.
Le deuxième programme utilise DRAW pour dessiner les losanges, puis les effacer.

POUR JOUER :
On commence l'exécution du programme avec RUN. On entre si nécessaire un noyau pour le générateur (premier programme), et on regarde.
On peut arrêter à tout moment en tapant deux fois ESC.

EXTENSIONS POSSIBLES :
On pourra introduire le son, utiliser l'instruction LINE qui permettra un tracé beaucoup plus rapide, imaginer sur des principes analogues divers programmes (kaléidoscope, tapisserie...).

```
10 REM LOSANGES1
20 RANDOMIZE(TIME)
30 MODE 0:INK 0,1:INK 1,0:INK 5,3:ORIGIN
 320,200
40 CLS:S=0
50 FOR K=0 TO 130 STEP 6
60 FOR I=0 TO K
70 IF S<>0 THEN ON INT(1*5*RND) GOTO 200
,210,220,230,240
80 PLOT -K+I,-I,1
90 PLOT -K+I,I,1
100 PLOT K-I,I,1
110 PLOT K-I,-I,1
120 NEXT I
130 NEXT K
140 FOR I=1 TO 4:C(I)=0:NEXT I
150 WHILE ((C(1)-C(2))*(C(1)-C(3))*(C(1)
-C(4))*(C(2)-C(3))*(C(2)-C(4))*(C(3)-C(4
)))=0
160 FOR I=1 TO 4:C(I)=2+INT(RND*14):NEXT
 I
170 WEND
180 S=NOT S
190 GOTO 50
200 PLOT -K+I,-I,C(1)
210 PLOT -K+I,I,C(2)
220 PLOT K-I,I,C(3)
230 PLOT K-I,-I,C(4)
240 GOTO 120
```

```
10 REM LOSANGES2
20 RANDOMIZE(TIME)
30 MODE 0:INK 0,1:INK 1,0:INK 5,3:ORIGIN
 320,200
40 CLS:S=0
50 FOR K=0 TO 130
70 IF S<>0 THEN ON INT(1*5*RND) GOTO 200
,210,220,230,240
80 MOVE 0,K
90 DRAW K,0,1
100 DRAW 0,-K,1
110 DRAW -K,0,1
120 DRAW 0,K,1
130 NEXT K
140 FOR I=1 TO 4:C(I)=0:NEXT I
150 WHILE ((C(1)-C(2))*(C(1)-C(3))*(C(1)
-C(4))*(C(2)-C(3))*(C(2)-C(4))*(C(3)-C(4
)))=0
160 FOR I=1 TO 4:C(I)=2+INT(RND*14):NEXT
 I
170 WEND
180 S=NOT S
190 GOTO 50
200 MOVE 0,K:DRAW K,0,C(1)
210 MOVE K,0:DRAW 0,-K,C(2)
220 MOVE 0,-K:DRAW -K,0,C(3)
230 MOVE -K,0:DRAW 0,K,C(4)
240 GOTO 120
```

Plus grand, plus petit

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction (un seul joueur).
Difficulté du jeu : difficile.

LE JEU :
Il s'inspire d'un principe légèrement différent de celui du Mastermind ; il s'agit également de deviner un nombre (de 1 à 7 chiffres), mais les indications dont on dispose sont cette fois d'une part le nombre de chiffres qui, dans le nombre essayé, sont supérieurs à ceux du nombre à deviner, et, d'autre part, le nombre de chiffres qui sont inférieurs. Si, par exemple, le nombre à deviner est 3Ø54, et que l'on essaie 4251, la machine répondra 2 plus grands (ce sont le 4 et le 2), et 1 plus petit (c'est le 1).

LE PROGRAMME :
La ligne 6Ø interdit de demander des nombres de plus de 8 chiffres.
Le nombre à deviner est placé en ligne 8Ø, dans une variable alphanumérique X$.
Les comparaisons avec le nombre essayé sont faites à la ligne 13Ø (qui teste si l'on a trouvé juste), et dans la boucle I (15Ø-18Ø) où l'on fait la comparaison "lettre à lettre".
L'affichage du résultat est réalisé grâce à la ligne 19Ø avec effacement d'un éventuel essai précédent.
Le compteur du nombre d'essais est en ligne 2ØØ.

POUR JOUER :
Le lancement du programme par RUN provoque l'affichage NOMBRE DE CHIFFRES.
On entre un nombre inférieur à 8 et on appuie sur la touche ENTER.
Le numéro de l'essai s'affiche, ainsi que la question NOMBRE ESSAYE ; on répond en entrant un nombre, suivi de ENTER.
Le résultat est alors affiché sous la forme > et <.
S'il n'est pas exact, on recommence (attention à entrer uniquement des nombres ayant autant de chiffres que le nombre à deviner).
Si l'on a trouvé, AMSTRAD affichera VOUS AVEZ TROUVE... EN... ESSAIS.
La ligne 24Ø forme une boucle d'attente. Puis AUTRE PARTIE (O/N) ; on fera O pour commencer une nouvelle partie.

EXTENSIONS POSSIBLES :
On pourra, comme pour le Mastermind, imaginer un jeu analogue avec des lettres, des couleurs, ou même en utilisant d'autres symbolismes.
On peut aussi prévoir de laisser jouer AMSTRAD qui tiendra alors le rôle de décodeur.

```
10 REM PLUS GRAND, PLUS PETIT
20 RANDOMIZE(TIME)
30 INK 0,22:BORDER 22:INK 1,1:INK 2,9
40 CLS
50 INPUT "NOMBRE DE CHIFFRES :";C
60 IF C<1 OR C>8 THEN PRINT CHR$(7):GOTO
 40
70 E=1:F=0
80 X$=STR$(INT(RND*9↑C+10↑(C-E)))
90 X$=RIGHT$(X$,LEN(X$)-1)
100 LOCATE 1,3:PRINT "ESSAI NO.  NOMBRE
ESSAYE     >        <"
110 LOCATE 1,4:PRINT STRING$(40,154)
120 LOCATE 1,F+5:PRINT TAB(4);CHR$(18);E
;TAB(18-(C/2)):INPUT "",P$
130 P$=LEFT$(P$,C):IF P$=X$ THEN 220
140 A=0:B=0
150 FOR I=1 TO C
160 IF MID$(P$,I,1)>MID$(X$,I,1) THEN A=
A+1
170 IF MID$(P$,I,1)<MID$(X$,I,1) THEN B=
B+1
180 NEXT I
190 LOCATE 28,F+5:PRINT A:LOCATE 36,F+5:
PRINT B
200 F=E MOD 17:E=E+1
210 GOTO 120
220 LOCATE 1,23:PRINT "VOUS AVEZ TROUVE
";:PEN 3:PRINT X$;:PEN 1:PRINT " EN";:PE
N 2:PRINT E;:PEN 1:PRINT "ESSAIS"
230 SOUND 1,478:SOUND 1,319:SOUND 1,379
240 X=TIME:WHILE TIME-X<1000:WEND
250 INPUT "AUTRE PARTIE (O/N)";R$
260 IF UPPER$(LEFT$(R$,1))="O" THEN 40
270 CLS
```

Ordre

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Les 1Ø premières lettres de l'alphabet sont affichées dans le désordre ; il faut réussir à les remettre dans l'ordre, avec le moins de coups possible, par des permutations successives de deux des lettres ; mais ceci a aussi pour inconvénient de faire permuter autour d'elles les deux lettres qui entourent celles-ci, suivant le mécanisme expliqué ci-dessous :

B A D G H I J C F E
0 1 2 3 4 5 6 7 8 9 : permutation 83

B A D F H I J E G C
Ø 1 2 3 4 5 6 7 8 9 : résultat

LE PROGRAMME :
Les boucles emboîtées I (6Ø-12Ø) et J (8Ø-1ØØ) définissent une variable alphanumérique A$, contenant les 1Ø premières lettres de l'alphabet dans le désordre.
La ligne 14Ø appelle le sous-programme 6ØØ qui affiche A$ avec des lettres multicolores. La ligne 15Ø affiche les numéros de lettres de l'alphabet pour la permutation.
On entre la permutation en ligne 18Ø.
Les lignes 19Ø et 2ØØ transforment la variable alphanumérique entrée en une variable numérique.
La ligne 21Ø appelle une première fois le sous-programme d'inversion.
La ligne 24Ø appelle une deuxième fois le sous-programme d'inversion.
La ligne 25Ø comptabilise les essais.
La ligne 17Ø contrôle les résultats.

POUR JOUER :
Lancer le programme par RUN.
Le mot initial est affiché, chaque lettre étant repérée par un numéro de 1 à 9.
A la question INVERSION (IJ), on répond en entrant un nombre de deux chiffres xy ; x est le numéro de la première lettre, et y le numéro de la seconde. Les deux inversions sont alors successivement effectuées, et l'on recommence.
Le nombre de coups sera affiché en fin de partie.

EXTENSIONS POSSIBLES :
On peut imaginer un jeu contre l'ordinateur, qui évaluerait le score du joueur en fonction de ses propres performances ; on peut aussi modifier les règles qui régissent les permutations.

```
10 REM ORDRE
20 RANDOMIZE(TIME)
30 MODE 0:FOR I=2 TO 11:INK I,I:NEXT I
40 CLS
50 A$=""
60 FOR I=1 TO 10
70 R$=CHR$(INT(RND*10+65))
80 FOR J=1 TO I-1
90 IF R$=MID$(A$,J,1) THEN 70
100 NEXT J
110 A$=A$+R$
120 NEXT I
130 S=1
140 LOCATE 1,4:PRINT"RESULTAT :";:GOSUB
600
150 PRINT TAB(11)"0123456789"
160 LOCATE 1,9:PRINT "ESSAI NUMERO";S
170 IF A$="ABCDEFGHIJ" THEN LOCATE 2,24:
PRINT "TROUVE EN";S;"COUPS":SOUND 1,478:
SOUND 1,319:SOUND 1,379:END
180 LOCATE 1,12:PRINT CHR$(20);:INPUT "I
NVERSION (IJ)";R$
190 I=ASC(LEFT$(R$,1))-47
200 J=ASC(RIGHT$(R$,1))-47
210 GOSUB 500
220 IF LEFT$(R$,1)="0" OR LEFT$(R$,1)="9
" THEN 240
230 I=I-1:J=I+2
240 GOSUB 500
250 S=S+1
260 GOTO 140
500 B$=""
510 N$=MID$(A$,I,1):M$=MID$(A$,J,1)
520 FOR T=1 TO LEN(A$)
530 IF T=I THEN B$=B$+M$:GOTO 560
540 IF T=J THEN B$=B$+N$:GOTO 560
550 B$=B$+MID$(A$,T,1)
560 NEXT T
570 A$=B$
580 RETURN
600 FOR T=1 TO 10
610 AF$=MID$(A$,T,1):PEN (ASC(AF$)-63):P
RINT AF$;
620 NEXT T
630 PEN 1:RETURN
```

Le pendu

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de lettres (version deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Il serait sans doute assez difficile de retrouver le nom de l'inventeur de ce jeu très à la mode, mais il est certain qu'il existe depuis des temps immémoriaux, quelquefois pratiqué sous des formes assez différentes (par exemple, reconstituer un petit cochonnet en lançant des dés).
L'apparition des jeux sur micro-ordinateur et la magie des écrans de télévision lui ont fait connaître un renouveau certain : dans la version retenue, l'un des joueurs doit entrer un mot de longueur quelconque. L'ordinateur affichera alors des tirets à l'emplacement des lettres de ce mot et demande au deuxième joueur de proposer une lettre. Chaque réponse correcte fait apparaître la lettre à son emplacement exact dans le mot, et chaque réponse fausse fait progresser le dessin de la potence.
Pour gagner, il suffit de réussir à trouver le mot inconnu avant d'être pendu.

LE PROGRAMME :
Le mot à deviner est introduit en début de programme (ligne 3Ø) ; des variables alphanumériques, en ligne 4Ø, contiennent tous les éléments du dessin en notation hexadécimale.
La boucle I (6Ø-8Ø) fabrique un mot P$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.
La boucle I (12Ø-18Ø) s'occupe du dessin (plus ou moins complet, selon la valeur de P). La boucle K (14Ø-16Ø) transforme les valeurs hexadécimales en éléments de dessin.
Les lignes 19Ø-2ØØ affichent le résultat et arrêtent le jeu.
La lettre essayée est entrée en ligne 21Ø et testée en ligne 23Ø ; le sous-programme 26Ø-28Ø remplace chaque tiret par la lettre correspondante.

POUR JOUER :
Après le lancement du programme par RUN passer en majuscules. L'un des joueurs doit entrer son mot et taper ENTER. Des tirets sont affichés à l'emplacement de chacune des lettres du mot ; et la question VOTRE LETTRE apparaît sur l'écran ; l'autre joueur répond alors en proposant une lettre suivie de ENTER.
Si la lettre est bonne, elle viendra remplacer un ou plusieurs tirets ; sinon. la potence va commencer à se dessiner.
Le mot exact sera affiché dans le cas où l'on aurait perdu.

EXTENSIONS POSSIBLES :
Il sera intéressant de constituer une liste de mots dans lesquels AMSTRAD pourra piocher au hasard ; on pourra aussi améliorer le dessin (en redéfinissant les caractères, par exemple) et modifier le nombre d'essais permis selon la difficulté du mot.

```
10 REM LE PÉNDU
20 CLS:INK 2,0,9
30 INPUT "VOTRE MOT :";R$
40 B$= "9B209B95209595209596 9B9C20952095
959595959595959596 9F9C20E120202020"
50 P$=""
60 FOR I=1 TO LEN(R$)
70 P$=P$+"-"
80 NEXT I
90 P=11
100 CLS
110 LOCATE 1,3:PRINT P$
120 FOR I=11 TO P STEP -1
130 PE$=""
140 FOR K=0 TO 2
150 PE$=PE$+CHR$(VAL("&"+MID$(B$,1+6*(I-
1)+2*K,2)))
160 NEXT K
170 PEN 3 :LOCATE 22,12-I:PRINT PE$;:PEN
 1
180 NEXT I
190 IF P$=R$ THEN PEN 2:LOCATE 1,12:PRIN
T "BRAVO!":SOUND 1,478:SOUND 1,319:SOUND
 1,379:GOTO 290
200 IF P=1 THEN INK 2,3,15:PEN 2:PRINT "
PERDU! LE MOT ETAIT :";R$:SOUND 1,478:GO
TO 290
210 LOCATE 1,20:PRINT CHR$(20);:INPUT "V
OTRE LETTRE :";S$:S$=LEFT$(UPPER$(S$+" "
),1)
220 T=0:I=0
230 I=INSTR(I+1,R$,S$): IF I<>0 THEN GOS
UB 260:GOTO 230
240 IF T=0 THEN P=P-1
250 GOTO 110
260 T=1
270 P$=LEFT$(P$,I-1)+S$+RIGHT$(P$,LEN(P$
)-I)
280 RETURN
290 GOTO 290
```

Belote

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de cartes.
Difficulté du jeu : facile.

LE JEU :
Le jeu proposé est plutôt une simulation, réalisée sur 32 cartes, de la distribution de 4 paquets de 5 cartes. Il pourra être utilisé pour l'étude de problèmes faisant intervenir des donnes de 2Ø cartes sur 32.

LE PROGRAMME :
En ligne 5Ø, on crée un tableau pouvant contenir les 2Ø cartes.
La boucle I (7Ø-13Ø) assure le tirage aléatoire de 2Ø nombres, pris parmi 32 ; la boucle J (1ØØ-12Ø) vérifie que l'on ne réalise pas deux fois le même tirage.
La deuxième boucle I (15Ø-29Ø) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 2ØØ à 26Ø.

POUR JOUER :
Il suffit de taper RUN pour obtenir immédiatement l'affichage de 4 paquets de 5 cartes chacun (une donne). Une double frappe ESC permet de reprendre la main.
Un nouveau RUN permet d'obtenir une nouvelle distribution.

EXTENSIONS POSSIBLES :
On pourra facilement mettre au point un affichage plus graphique en redéfinissant les caractères de l'AMSTRAD ; on pourra également se servir de ce programme comme d'un noyau pour la mise au point de programmes de jeux de cartes sur l'ordinateur (qui pourra, si nécessaire, servir de partenaire : voir l'exemple du Poker, dans le même ouvrage).

```
10 REM BELOTE
20 RANDOMIZE(TIME)
30 INK 0,9:INK 2,0:INK 3,3
40 CLS
50 DIM A(20)
60 LOCATE 12,12:PRINT "PATIENTEZ, JE TIR
E"
70 FOR I=1 TO 20
80 A(I)=1+INT(RND*32)
90 IF I=1 THEN 130
100 FOR J=1 TO I-1
110 IF A(I)=A(J) THEN 80
120 NEXT J
130 NEXT I
140 CLS
150 FOR I=1 TO 20
160 S$=CHR$(227)
170 S=1+A(I) MOD 8
180 R$=STR$(S+6)+"  ":R$=RIGHT$(R$,LEN(R
$)-1)
190 R$=LEFT$(R$,2)
200 IF S=5 THEN R$="VA"
210 IF S=6 THEN R$="DA"
220 IF S=7 THEN R$="RO"
230 IF S=8 THEN R$="AS"
240 IF A(I)<=24 THEN S$=CHR$(226)
250 IF A(I)<=16 THEN S$=CHR$(229)
260 IF A(I)<=8 THEN S$=CHR$(228)
270 LOCATE 3+10*INT((I-1)/5),2*K+1:PEN (
3+((ASC(S$) MOD 3)=1)):PRINT R$;"  ";S$
280 K=(K+1) MOD 5
290 NEXT I
300 GOTO 300
```

Un taquin de calcul

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

LE JEU :
Il se pratique à deux joueurs, l'un tenant le rôle du codeur, l'autre celui du décodeur.
Le codeur part d'un nombre (proposé par AMSTRAD) et doit le transformer à l'aide de 5 opérations qu'il choisit lui-même, ainsi d'ailleurs que les opérateurs.
Le résultat qu'il a obtenu et les opérations qu'il a utilisées seront affichés sur l'écran : le décodeur devra alors essayer de retrouver le nombre initial en effectuant les opérations en sens inverse.

LE PROGRAMME :
La ligne 5Ø permet d'obtenir l'entier de départ compris entre 5 et 2Ø.
Une boucle I (11Ø-18Ø) permet d'entrer successivement les 5 opérations.
4 sous-programmes sont utilisés pour les calculs (aux lignes 1ØØØ et 4ØØØ).
Une boucle WHILE/WEND (12Ø-14Ø) assure la validité des opérations.
Après chaque calcul, le résultat est affiché (seul le codeur devra regarder l'écran pendant cette opération) et le résultat définitif restera affiché pour le décodeur à la fin des calculs (ligne 17Ø).
La ligne 19Ø affiche les 5 opérations utilisées, ainsi que les opérateurs correspondants.

POUR JOUER :
On frappe RUN pour lancer le programme.
Le nombre du début est affiché, puis la question ENTRER 5 OPERATIONS ; on devra les entrer l'une après l'autre, sous la forme, par exemple : + 4 suivi de ENTER.
Le résultat sera affiché après chaque opération entrée. A la fin, les opérations entrées sont toutes affichées sur une même ligne, et le message BON COURAGE est envoyé à l'adresse du décodeur (il n'y a pas de contrôle au décodage).

EXTENSIONS POSSIBLES :
On pourra, bien sûr, songer à améliorer la partie décodage du programme en introduisant des contrôles sur le résultat ; on peut aussi envisager de laisser AMSTRAD participer davantage en lui confiant le rôle du codage, etc.

```
10 REM UN TAQUIN DE CALCUL
20 RANDOMIZE(TIME)
30 INK 0,3:INK 3,18,2
40 CLS
50 X=INT(RND*16+5)
60 PRINT "NOMBRE INITIAL :";X
70 PRINT "ENTRER 5 OPERATIONS"
80 PRINT "SOUS LA FORME +5, /2 ,*4 ..."
90 F=X
100 DIM R$(5)
110 FOR I=1 TO 5
120 WHILE INSTR("*+-/",LEFT$(R$(I)+" ",1
))=0
130 LOCATE 1,6:PRINT CHR$(18);:INPUT "VO
TRE OPERATION :";R$(I)
140 WEND
150 A=INSTR("*+-/",LEFT$(R$(I),1))
160 ON A GOSUB 1000,2000,3000,4000
170 LOCATE 1,9:PRINT "NOMBRE OBTENU   :"
;CHR$(20);F
180 NEXT I
190 PRINT:PRINT:PRINT "OPERATIONS :";:PE
N 2:PRINT R$(4);" ";R$(1);" ";R$(5);" ";
R$(3);" ";R$(2)
200 PEN 3:PRINT:PRINT:PRINT "BON COURAGE
 ...."
210 SOUND 1,478:SOUND 1,319:SOUND 1,379:
PEN 1:END
1000 F= F*VAL(RIGHT$(R$(I),LEN(R$(I))-1)
)
1010 RETURN
2000 F= F+VAL(RIGHT$(R$(I),LEN(R$(I))-1)
)
2010 RETURN
3000 F= F-VAL(RIGHT$(R$(I),LEN(R$(I))-1)
)
3010 RETURN
4000 F= F/VAL(RIGHT$(R$(I),LEN(R$(I))-1)
)
4010 RETURN
```

Pendu inversé

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu sur les mots.
Difficulté du jeu : facile.

LE JEU :
Le Pendu est un jeu de lettres très connu, présenté par ailleurs dans ce recueil, dans lequel il s'agit de deviner un mot caché par l'ordinateur et où chaque lettre non découverte vous rapproche un peu plus de la potence. Ici, c'est AMSTRAD qui va chercher à deviner votre mot, mais il joue quelque peu au hasard et ne possède pas votre intuition ; il faudra donc, en règle générale, beaucoup d'essais pour découvrir la solution. Pour augmenter ses chances, on a fait appel à une formule qui lui permet de tirer davantage dans les voyelles puis, dans les consonnes courantes, et, enfin, dans les consonnes plus rares.

LE PROGRAMME :
La ligne 4Ø permet d'enregistrer, dans une variable alphanumérique D$, les lettres de l'alphabet, dans l'ordre : voyelles, consonnes usuelles, autres consonnes. Le mot à deviner est introduit en ligne 6Ø.
La ligne 9Ø imprime des tirets aux emplacements des lettres à trouver.
La boucle WHILE/WEND (1ØØ-22Ø) assure la continuité de la recherche.
La boucle en ligne 12Ø sert uniquement à simuler le temps de recherche d'une lettre.
La formule, placée en ligne 13Ø, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.
La ligne 15Ø provoque un nouveau tirage, et la boucle I (18Ø-2ØØ) réimprime les tirets et les lettres trouvées.
La ligne 21Ø incrémente le compteur d'essais.
La ligne 23Ø affiche le nombre d'essais réalisés, en fin de partie.

POUR JOUER :
Lancer le programme par RUN et passer en majuscules.
Entrer le mot à découvrir, suivi de ENTER.
Des tirets sont alors affichés (un par lettre) et AMSTRAD propose une première lettre, regarde si elle convient et continue automatiquement, jusqu'à ce qu'il ait trouvé.
On pourra sortir du programme à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
Il sera sans doute intéressant, pour de jeunes enfants, de prévoir une version non automatique où le joueur répondrait OUI ou NON pour chaque lettre proposée par AMSTRAD ; on pourra aussi imaginer d'ajouter un affichage graphique, par exemple une potence ou une voiture qui avance vers un précipice, etc.

```
10 REM PENDU INVERSE
20 RANDOMIZE(TIME)
30 CLS
40 D$="YAEIOUBCDFGLMNPRSTVWHKJQXZ"
50 C=1:T$=""
60 INPUT "ENTREZ LE MOT :";R$
70 CLS
80 T$=STRING$(LEN(R$),"-")
90 LOCATE 1,6:PEN 3:PRINT T$:PEN 1
100 WHILE T$<>R$
110 LOCATE 1,9:PRINT "JE CHERCHE .......
  ESSAI";C
120 FOR J=1 TO 1000:NEXT J
130 X=-6*(RND<0.7)-13*(RND<0.5)-7*(RND<0
.4)
140 X=INT(X*RND+1)
150 E$=MID$(D$,X,1):IF E$="-" THEN 130
160 D$=LEFT$(D$,X-1)+"-"+RIGHT$(D$,26-X)
170 PRINT:PRINT:PRINT "JE PRENDS :";:PEN
 2:PRINT E$:SOUND 1,239:SOUND 1,213:SOUN
D 1,190:PEN 1
180 FOR I=1 TO LEN(R$)
190 IF E$=MID$(R$,I,1) THEN T$=LEFT$(T$,
I-1)+E$+RIGHT$(T$,LEN(T$)-I):PEN 2:LOCAT
E I,6:PRINT MID$(T$,I,1):PEN 1
200 NEXT I
210 C=C+1
220 WEND
230 LOCATE 11,20:PAPER 3:PEN 2:PRINT "TR
OUVE EN";C-1;"COUPS"
240 SOUND 1,478:SOUND 1,319:SOUND 1,379
250 GOTO 250
```

Nicomaque

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion (calcul mental).
Difficulté du jeu : moyenne.

LE JEU :
Il utilise une propriété un peu particulière des entiers de 1 à 1ØØ qui peuvent être identifiés sans ambiguïté par leurs restes dans les divisions par 3, 5 et 7.
Le programme, proposé ici, est un programme codeur qui affiche simplement les restes d'un entier donné N dans les divisions indiquées.
Le jeu consiste à essayer de découvrir l'entier (mais, dans cette version, vous n'avez droit qu'à un seul essai).

LE PROGRAMME :
La ligne 5Ø permet de confectionner l'entier.
La boucle I (8Ø-11Ø) calcule les trois restes successifs, et la ligne 1ØØ permet de les afficher sous une forme assez visuelle, aux bons emplacements.
Les lignes 6Ø et 7Ø sont là pour indiquer de quelle division il s'agit.
Le nombre proposé par le joueur est entré à la ligne 12Ø ; il est affiché et testé en ligne 13Ø (affichage de fin de partie : BRAVO ou PERDU).
La forme d'enveloppe sonore définie en ligne 15Ø permet de jouer les notes (SOUND) de façon déliée en ligne 16Ø.
Les lignes 17Ø et 18Ø s'occupent de gérer la prochaine partie, selon la réponse du joueur.

POUR JOUER :
On lance le programme en tapant RUN.
Les trois restes seront alors affichés sur la même ligne, symbolisés par des étoiles (l'absence de carrés signifiant : reste Ø).
A la question VOTRE ESSAI, on répondra en donnant un entier compris entre Ø et 1ØØ, suivi de ENTER.
On obtiendra, en réponse, un affichage PERDU ou BRAVO, puis la question UNE AUTRE PARTIE (O/N).
Pour faire un nouvel essai, on répondra O, ou N suivi de ENTER.

EXTENSIONS POSSIBLES :
On peut envisager de réaliser le programme décodeur (on donne les restes et AMSTRAD devra vous donner N) et chercher si la propriété peut être étendue (en effectuant davantage de divisions, par exemple).

```
10 REM NICOMAQUE
20 RANDOMIZE(TIME)
30 ZONE 8
40 CLS
50 X=INT(RND*100+1)
60 PRINT TAB(6)"RESTE DANS LA DIVISION P
AR"
70 PRINT:PRINT ,"3","5","7"
80 FOR I=1 TO 3
90 A=X MOD (2*I+1)
100 PAPER 3:PEN 0:LOCATE 8*I,5:PRINT STR
ING$(A,233)
110 NEXT I
120 PAPER 0:PEN 1:LOCATE 1,10:INPUT "VOT
RE ESSAI :",N
130 PAPER 2:PEN 3:PRINT:IF N=X THEN PRIN
T "BRAVO" ELSE PRINT "PERDU!"
140 PAPER 0:PEN 1
150 ENV 1,1,15,10,1,-15,10
160 FOR I=1 TO 3:SOUND 1,413,20,0,1:NEXT
 I
170 PRINT:INPUT "UNE AUTRE PARTIE (O/N)"
,R$
180 IF LEFT$(UPPER$(R$+" "),1)="O" THEN
40
```

Mastermind

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : logique (un seul joueur).
Difficulté du jeu : difficile.

LE JEU :
Devenu un grand classique des jeux sur micro-ordinateur, ce jeu développe le raisonnement déductif ; il faut réussir à trouver le plus vite possible un nombre de n chiffres (on pourra prendre n compris entre 1 et 6), en disposant à chaque essai de renseignements.
En effet, AMSTRAD répondra en indiquant d'une part le nombre de chiffres qui sont, dans le nombre proposé, exactement à la même place que dans le nombre à découvrir, et, d'autre part, le nombre de chiffres qui figurent dans le nombre à découvrir sans être exactement au bon emplacement.

LE PROGRAMME :
Le choix du nombre de chiffres du nombre à découvrir est effectué aux lignes 4Ø à 6Ø.
Le nombre est fabriqué à la ligne 8Ø (il pourra y avoir des chiffres répétés).
L'affichage est préparé par la ligne 1ØØ.
La boucle WHILE/WEND (12Ø-24Ø) s'occupe de la fin de partie.
Une première boucle I (16Ø-21Ø) teste les chiffres qui sont à la bonne place.
Une deuxième boucle J (18Ø-2ØØ) teste les chiffres qui sont présents et mal placés.
Les compteurs sont aux lignes 17Ø et 19Ø, tandis que l'emploi de la variable EB$ permet d'éviter les comptages en double.
La ligne 22Ø affiche le résultat, et la ligne 23Ø comptabilise les essais.

POUR JOUER :
On tape RUN et l'on choisit la longueur du nombre désiré (de 1 à 6), suivie de ENTER.
Après l'affichage NO. NOMBRE BON PLACE, on introduit son propre essai, suivi de ENTER.
On obtient immédiatement la réponse (en cas de double chiffre, les résultats peuvent être un peu délicats à interpréter).
On introduit ensuite le nouvel essai, et ainsi de suite.
Lorsque l'on a trouvé, on obtiendra l'affichage TROUVE EN ... COUPS.
On peut sortir du programme à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
On peut chercher à éviter la présence de chiffres doubles dans le nombre à trouver, on peut aussi prévoir un changement de page automatique, avec rappel des résultats précédents ; on pourra aussi réaliser des jeux analogues avec des mots, ou d'autres symboles (couleur, taille...).

```
10 REM MASTERMIND
20 DEFINT N
30 RANDOMIZE(TIME)
40 WHILE N<1 OR N>6
50 INPUT "COMBIEN DE CHIFFRES :";N
60 WEND
70 E=1:F=1:X$=STR$(INT(RND*9↑N+10↑(N-1))
)
80 X$=RIGHT$(X$,LEN(X$)-1)
90 CLS
100 PRINT "  NO.        NOMBRE      BON
  PLACE"
110 PRINT STRING$(40,154)
120 WHILE E$<>X$
130 LOCATE 1,3+F:PRINT E;CHR$(18);TAB(13
);:INPUT"",E$
140 X=0:Y=X
150 EB$=E$
160 FOR I=1 TO N
170 IF MID$(X$,I,1)=MID$(E$,I,1) THEN X=
X+1:MID$(EB$,I,1)=".":GOTO 210
180 FOR J=1 TO N
190 IF MID$(X$,I,1)=MID$(EB$,J,1) THEN Y
=Y+1:MID$(EB$,J,1)=".":GOTO 210
200 NEXT J
210 NEXT I
220 LOCATE 24,3+F:PEN 2:PRINT X;TAB(33);
:PEN 3:PRINT Y:PEN 1
230 E=E+1:F=(F+1) MOD 18
240 WEND
250 SOUND 1,478:SOUND 1,319:SOUND 1,379
260 LOCATE 11-(N/2),22:PAPER 3:PRINT "TR
OUVE ";E$;" EN";E-1;"COUPS"
270 GOTO 270
```

Idem

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de calcul mental (2 joueurs).
Difficulté du jeu : selon le niveau choisi.

LE JEU :
AMSTRAD choisit un nombre de n chiffres (selon le niveau de difficulté indiqué au début). Ce nombre (le même au début pour les 2 joueurs) sera régulièrement décrémenté à chaque tour, d'une quantité également choisie au hasard par AMSTRAD en début de jeu ; vous devez, à chaque coup, choisir une colonne (les dizaines, les centaines, etc.) et ajouter dans cette colonne un certain entier ; le but est d'arriver à obtenir à un moment l'affichage d'un nombre ayant tous ses chiffres identiques.
Chaque résultat sera affiché un court instant, après quoi la main passera au joueur suivant, qui repartira de son précédent résultat.

LE PROGRAMME :
Une variable numérique N est dimensionnée à 2 en ligne 4Ø, pour contenir les résultats successifs des deux joueurs.
La boucle WHILE/WEND (45-55) permettra d'entrer le niveau de difficulté choisi.
Le nombre initial est confectionné par la ligne 7Ø. La quantité qui interviendra dans les décrémentations est obtenue en ligne 9Ø.
La ligne 12Ø teste si le résultat n'est pas trop petit.
La ligne 13Ø affiche le numéro du joueur.
Le nombre est affiché en ligne 14Ø et le jeu du joueur est entré en ligne 16Ø (nombre de deux chiffres).
La ligne 17Ø incrémente la colonne choisie, et la ligne 18Ø décrémente le nombre de la quantité choisie au départ.
Le résultat sera affiché un court instant (ligne 21Ø). La boucle I (26Ø-28Ø) permet de vérifier si tous les chiffres sont identiques.
L'impression et la fin de partie sont effectuées en ligne 29Ø. Les lignes 3ØØ et 31Ø s'occupent du changement de joueur.

POUR JOUER :
Lancer l'exécution du programme par RUN.
Le nombre initial sera affiché pour le joueur n° 1, qui joue en tapant xy, suivi de ENTER ; x est le chiffre à ajouter, et y est le numéro de la colonne où il faut l'ajouter.
Le résultat (provisoire s'il est trop petit) sera alors affiché un court instant, et le trait passe au deuxième joueur, et ainsi de suite. On peut arrêter à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
On pourra étendre le jeu à plus de 2 joueurs, améliorer le choix du nombre intervenant dans la décrémentation, améliorer les divers contrôles, laisser les différents résultats affichés, etc.

```
10 REM IDEM
20 RANDOMIZE(TIME)
40 CLS:H=1:DIM N(2)
45 WHILE N<1 OR N>8
50 INPUT "NIVEAU (2-7) :";N
55 WEND
60 U=10:P=0
70 N(1)=INT((U↑N-U↑(N-1))*RND+U↑(N-1))
80 N(2)=N(1)
90 A=INT(2*U↑(N-1)*RND+U↑(N-1))
100 IF H=1 THEN P=P+1
110 CLS
120 IF N(H)<U↑(N-1) THEN N(H)=U↑N-ABS(N(
H))
130 LOCATE 1,3:PRINT "JOUEUR";:PEN 1+H:P
RINT H;:PEN 1:PRINT "   ESSAI";P
140 LOCATE 1,6:PRINT "NOMBRE :":LOCATE 1
5-N,6:PEN 1+H:PRINT N(H):PEN 1
150 LOCATE 9,7:PRINT "7654321"
160 LOCATE 1,10:PRINT "VOTRE JEU (N ET C
) SOUS FORME NC ";:PEN H+1:INPUT "",R$:P
EN 1
170 N(H)=N(H)+VAL(LEFT$(R$,1))*U↑(VAL(RI
GHT$(R$,1))-1)
180 N(H)=N(H)-A
190 IF N(H)>U↑N THEN N(H)=N(H)-U↑N
200 N(H)=INT(N(H))
210 LOCATE 1,13:PRINT "RESULTAT :";N(H)
220 SOUND 1,478:SOUND 1,426:SOUND 1,379
230 FOR J=1 TO 1200:NEXT J
240 R$=STR$(N(H))
250 R$=RIGHT$(R$,LEN(R$)-1)
260 FOR I=1 TO LEN(R$)-1
270 IF MID$(R$,I,1)<>MID$(R$,I+1,1) THEN
 300
280 NEXT I
290 PRINT:PRINT:PEN H+1:PRINT "BRAVO! JO
UEUR";H:SOUND 1,478:SOUND 1,319:SOUND 1,
379:END
300 H=1+(H MOD 2)
310 GOTO 100
```

Reines sur l'échiquier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Le programme utilise un algorithme très performant pour résoudre un problème qui a donné du fil à retordre à des chercheurs, même aussi célèbres que le mathématicien GAUSS, qui n'avait pas pu découvrir toutes les solutions. Il s'agit de trouver toutes les manières de disposer des reines sur un échiquier, de telle sorte qu'aucune ne puisse en prendre une autre. Le programme proposé est très souple, puisque vous disposez même de la possibilité de choisir la taille de votre échiquier, et les résultats seront affichés à la fois sous forme visuelle et sous forme numérique.

LE PROGRAMME :
La ligne 3Ø vous permet de choisir la taille de l'échiquier.
Une variable D sera dimensionnée en conséquence, en ligne 4Ø.
Une boucle d'exploration principale I (5Ø-11Ø) contient une sous-boucle J (8Ø-1ØØ) qui réalise le contrôle de non-alignement (ligne 9Ø).
Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 23Ø à 27Ø et on recommence.
La boucle F (13Ø-21Ø) assure l'impression de l'échiquier, et la boucle K (16Ø-18Ø), que l'on pourra éventuellement supprimer, assure l'impression du résultat sous forme d'un code numérique correspondant à la position des pièces dans chaque colonne.
La ligne 22Ø permet un arrêt après l'affichage de chacune des solutions.

POUR JOUER :
On lance le programme avec RUN, puis on tape la dimension de l'échiquier désirée, et on fait ENTER.
Il suffit d'attendre, avec patience (plus l'échiquier sera grand, plus l'attente sera longue). La première solution sera affichée puis le programme s'arrête, et il faut appuyer sur une touche quelconque pour le faire repartir. Pour terminer l'exploration sans avoir eu toutes les solutions, il faudra taper ESC deux fois.

EXTENSIONS POSSIBLES :
On peut envisager de trouver des algorithmes pour tous les problèmes analogues (position de tours, déplacement d'un cavalier, etc.), en essayant d'utiliser un maximum de boucles WHILE/WEND.

```
10 REM REINES SUR L'ECHIQUIER
20 CLS:INK 0,9:INK 1,21:INK 2,0:INK 3,26
,15:SPEED INK 5,5
30 INPUT "TAILLE DE L'ECHIQUIER :",S
40 DIM D(S)
50 I=I+1
60 D(I)=1
70 IF I=1 THEN 110
80 FOR J=1 TO I-1
90 IF D(I)=D(J) OR (ABS(D(I)-D(J))=I-J)
THEN 240
100 NEXT J
110 I=I+1:IF I<=S THEN 60
120 CLS
130 FOR F=1 TO S
140 LOCATE 2*F+1,1:PRINT F
150 LOCATE 1,2*F+1:PRINT F
160 FOR K=1 TO S
170 PEN 2:LOCATE 2*K+2,2*F+1:PRINT ".":P
EN 1
180 NEXT K
190 LOCATE 2*D(F)+2,2*F+1:PEN 3:PRINT "*
":PEN 1
200 LOCATE F+1,20:PEN 3:PRINT RIGHT$(STR
$(D(F)),1):PEN 1
210 NEXT F
220 SOUND 1,478:SOUND 1,319:SOUND 1,379:
LOCATE 6,23:INPUT "FAIRE <ENTER> POUR CO
NTINUER ",R$
230 I=I-1
240 D(I)=D(I)+1
250 IF D(I)<=S THEN 70
260 I=I-1
270 IF I<>0 THEN 240
```

Mots

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de lettres (pour 1 ou plusieurs joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Il est difficile d'imaginer plus simple dans son principe que ce jeu, qui se contente d'afficher dans le désordre les lettres prises dans une trentaine de mots. Mais essayez de jouer, et vous verrez que ce n'est pas si facile, la difficulté variant évidemment selon la liste de mots que vous aurez introduits en DATA au préalable.

LE PROGRAMME :
La ligne 4Ø permet de dimensionner la variable A$ (I) qui contiendra les mots.
La boucle I (5Ø) saisit la liste des mots (inscrits en DATA aux lignes 24Ø à 28Ø).
La boucle 7Ø-9Ø choisit trois couleurs d'encre différentes entre elles et différentes de celle du fond.
La ligne 12Ø choisit un mot au hasard ; la boucle I (15Ø-22Ø) contrôle si une lettre n'a pas déjà été tirée, et reprend les lettres dans le désordre.
La ligne 23Ø affiche les lettres, et la ligne 24Ø attend votre proposition, qui est testée en ligne 25Ø.
Le nombre d'essais est incrémenté en ligne 26Ø.
La fin de partie est contrôlée par la ligne 27Ø, et le score est affiché en ligne 29Ø.

POUR JOUER :
On lance le programme par RUN. Une série de lettres est affichée dans le désordre, et il faut essayer de retrouver le mot correspondant (le temps est contrôlé).
On tape le mot supposé, puis ENTER, et une nouvelle série de lettres sera affichée, si l'on répond O à la question AUTRE ESSAI (O/N).
Le score et le nombre d'essais sont affichés en fin de partie.

EXTENSIONS POSSIBLES :
On peut imaginer de minuter le temps d'apparition du mot, ou de faire défiler les lettres l'une après l'autre sur l'écran.
On peut aussi corser la difficulté en introduisant des JOKERS, jouer à plusieurs, tenir le compte des points, etc.

```
10 REM MOTS
20 RANDOMIZE(TIME)
30 DEFINT E
40 DIM A$(30):S=0:E=0
50 FOR I=1 TO 30:READ A$(I):NEXT I
60 F=0:E1=0:E2=0:E3=0
70 WHILE ((F-E1)*(F-E2)*(F-E3)*(E1-E2)*(
E1-E3)*(E2-E3))=0
80 E1=RND*26:E2=RND*26:E3=RND*26
90 WEND
100 CLS:INK 0,F:INK 1,E1:INK 2,E2:INK 3,
E3
110 WHILE  A$(Z)=""
120 Z=INT(RND*30+1)
130 WEND
140 R$=A$(Z):A$="":A$(Z)="":V$=R$
150 FOR I=1 TO LEN(R$)
160 Z=1
170 WHILE MID$(R$,Z,1)="-"
180 Z=INT(RND*LEN(R$)+1)
190 WEND
200 A$=A$+MID$(R$,Z,1)
210 MID$(R$,Z,1)="-"
220 NEXT I
230 PRINT "LES LETTRES SONT :";:PEN 3:PR
INT A$:SOUND 1,478:SOUND 1,426:SOUND 1,3
79:PEN 1
240 LOCATE 1,6:PRINT "VOTRE SOLUTION   :
";:PEN 2:INPUT "",S$:PEN 1
250 IF UPPER$(S$)=V$ THEN S=S+1
260 E=E+1
270 LOCATE 1,11:INPUT "AUTRE ESSAI (O/N)
:",R$
280 IF LEFT$(UPPER$(R$+" "),1)="O" AND E
<30 THEN 60
290 LOCATE 1,16:PRINT "VOTRE SCORE :";S;
"EN";E;"ESSAIS"
300 SOUND 2,478:SOUND 2,319:SOUND 2,379
310 DATA JAMBON,ENERGUMENE,AFFILIE,ERUDI
T,INSIFIGNIANT,FATALISTE
320 DATA IMPRESSIONNANT,ALLECHANT,ANTICO
NSTITUTIONNEL,EPEIRE,INTREPIDE,ORDINATEU
R
330 DATA TREVE,BANDEROLE,SENATEUR,ACTUAL
ITE,CANNIBALE,DIEDRE
340 DATA IVRE,BATEAU,PLUMEAU,SIGNATAIRE,
CONTRACTUEL,PROCES
350 DATA CHALEUR,HIVERNAL,DESODORISANT,L
UXATION,EBONITE,NOVICE
```

Permutations

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.

LE JEU :
AMSTRAD vous demande d'abord le nombre de lettres que vous désirez permuter (pas plus de 12). A partir de là, il va vous afficher le nombre correspondant de lettres prises dans le début de l'alphabet, d'abord dans le désordre (DEBUT) puis dans l'ordre (FIN).
Vous devez réussir à remettre dans l'ordre les lettres données au début, en utilisant des permutations sur les n premières lettres à partir de la gauche, avec le moins d'essais possible.

LE PROGRAMME :
La ligne 4Ø permet de choisir le nombre de lettres, et la ligne 5Ø contrôle que l'on n'en demande pas plus de 12.
La ligne 8Ø réalise le brouillage, en tirant au hasard la nouvelle position de chaque lettre. Les lettres sont ensuite ajoutées au fur et à mesure de leur tirage, dans une variable B$ (ligne 1ØØ), supprimées dans A$ et replacées en C$, dans le bon ordre (lignes 11Ø-12Ø).
La ligne 14Ø contrôle le nombre de lettres de B$.
Les affichages sont effectués aux lignes 16Ø, 17Ø et 18Ø.
Le retournement est fait, lettre par lettre, dans la boucle I (21Ø-23Ø) ; le résultat est placé dans B$ en ligne 24Ø, affiché par la ligne 25Ø et comparé au résultat final en ligne 26Ø.
Le nombre d'essais sera affiché en fin de partie, grâce à la ligne 27Ø.

POUR JOUER :
Après avoir tapé RUN, on obtient l'affichage de la question NOMBRE DE LETTRES (DE 1 A 12) ; on répond en frappant N, suivi de ENTER. L'affichage suivant présente l'ordre initial (DEBUT) et l'ordre à obtenir (FIN).
On joue en indiquant le nombre de lettres à permuter à partir de la gauche, par exemple : sur un affichage initial de GEABDFC, si l'on tape 3 et ENTER, on obtiendra AEGBDFC.
On poursuivra jusqu'à l'obtention de l'ordre désiré.
Le score sera affiché en fin de partie ainsi qu'une question AUTRE PARTIE (O/N) ; répondre O si l'on désire faire une autre partie.

EXTENSIONS POSSIBLES :
On pourra utiliser le même type de jeu pour, par exemple, reconstituer un dessin caché dont les diverses parties seraient repérées par des lettres. On peut aussi réaliser des permutations plus complexes (par exemple le fait de permuter deux lettres permute les deux voisines de la lettre la plus à gauche, etc.).

```
10 REM PERMUTATIONS
20 RANDOMIZE(TIME)
30 CLS
40 INPUT "NOMBRE DE LETTRES (DE 1 A 12)
:";N
50 IF N<2 OR N>12 THEN 40
60 A$="ABCDEFGHIJKL":C$=STRING$(12," "):
B$=""
70 CLS
80 X=INT(RND*N+1)
90 IF MID$(A$,X,1)=" " THEN 80
100 B$=MID$(A$,X,1)+B$
110 MID$(C$,X,1)=MID$(A$,X,1)
120 MID$(A$,X,1)=" "
130 S=1
140 IF LEN(B$)<>N THEN 80
150 C$=LEFT$(C$,LEN(B$))
160 LOCATE 1,5:PRINT "DEBUT     :";:PEN 2
:PRINT B$:PEN 1
170 LOCATE 1,7:PRINT "FIN       :";:PEN 3
:PRINT C$:PEN 1
180 LOCATE 1,11:PRINT "PERMUTATION NUMER
O";S:PRINT:PRINT "VOTRE QUANTITE ";CHR$(
18);:INPUT "",L:IF L>N OR L<1 THEN 180
190 S=S+1
200 A$=""
210 FOR I=1 TO L
220 A$=A$+MID$(B$,L-I+1,1)
230 NEXT I
240 B$=A$+RIGHT$(B$,LEN(B$)-L)
250 SOUND 1,478:SOUND 1,426:SOUND 1,379:
LOCATE 1,17:PRINT "RESULTAT :";:PAPER 2:
PEN 3:PRINT B$:PAPER 0:PEN 1
260 IF B$<>C$ THEN 180
270 SOUND 1,478:SOUND 1,319:SOUND 1,379:
PRINT:PRINT "TROUVE EN";S-1;"ESSAIS"
280 INPUT "AUTRE PARTIE (O/N)";R$
290 IF LEFT$(R$,1)="O" THEN 30
```

Bridge

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : variable, selon le problème posé.

LE JEU :
Vous ne trouverez pas, dans ce programme, un partenaire pour vos bridges mais, plus exactement, une simulation pour l'affichage d'une donne ; cet affichage est réalisé jeu par jeu. On peut, à partir de là, se poser toutes sortes de problèmes : annonces, jeu de la carte, etc.

LE PROGRAMME :
Les lignes 5Ø et 6Ø définissent un jeu de 52 cartes.
La table B$, en ligne 7Ø, permettra d'afficher les symboles des couleurs et aussi d'améliorer la désignation de certaines cartes (1Ø, valet, dame, roi et as).
La boucle J (8Ø-24Ø) réalise le tirage de quatre jeux de 13 cartes.
La boucle I (9Ø-21Ø) réalise tirage et impression.
La carte est choisie, parmi les 52 disponibles, par la boucle WHILE/WEND (11Ø-13Ø) initialisée en 1ØØ.
La ligne 14Ø calcule la couleur de la carte tirée.
Les lignes 15Ø et 16Ø s'occupent de l'impression complémentaire dans le cas où l'on tire un 1Ø, un valet, une dame, un roi ou un as.
L'impression est réalisée par les lignes 17Ø à 19Ø.
La ligne 2ØØ permet d'éliminer les cartes déjà tirées, et la ligne 22Ø réalise un arrêt entre chaque affichage.

POUR JOUER :
On lance le programme par RUN.
Une première série de 13 cartes est affichée verticalement. Il suffira d'appuyer sur une touche quelconque pour obtenir la deuxième série de 13 cartes. On procédera de même pour la troisième et la quatrième.

EXTENSIONS POSSIBLES :
On peut imaginer de représenter les 13 cartes sous une forme plus conventionnelle.
On peut aussi ajouter la possibilité de faire des annonces (et de les comparer à celles de l'AMSTRAD, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour AMSTRAD.

```
10 REM BRIDGE
20 RANDOMIZE(TIME)
30 INK 2,0:INK 3,6
40 CLS
50 R$="234567891VDRA"
60 R$=R$+R$+R$+R$+"O"
70 B$=CHR$(226)+CHR$(227)+CHR$(228)+CHR$
(229)+"OAAOS"
80 FOR J=1 TO 4
90 FOR I=1 TO 13
100 X=53
110 WHILE MID$(R$,X,1)="O"
120 X=INT(RND*LEN(R$)+1)
130 WEND
140 S=INT((X-1)/13)+1
150 T=X MOD 13
160 T=T-13*(T=0)
170 LOCATE 10*J-9,I+1:PRINT MID$(R$,X,1)
;
180 IF T>=9 THEN PRINT MID$(B$,T-4,1)
190 LOCATE 10*J-5,I+1:PEN 2-(S=2 OR S=3)
:PRINT MID$(B$,S,1):PEN 1
200 MID$(R$,X,1)="O"
210 NEXT I
220 LOCATE 4,23:INPUT "FAIRE <ENTER> POU
R CONTINUER";S$
230 LOCATE 4,23:PRINT CHR$(20);
240 NEXT J
```

Hanoï

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il est présenté, ici, dans une version pour un joueur. On dispose d'un certain nombre de disques (7 au plus) et l'on doit les transférer de la tour numéro 1 à la tour numéro 3, en respectant une seule règle : à aucun moment, un disque ne peut être recouvert par un disque de diamètre supérieur.
Le nombre de coups n'est pas comptabilisé par ce programme mais, par contre, les coups illicites seront refusés.

LE PROGRAMME :
La variable D$, en ligne 4Ø, servira à dessiner les disques, et les socles.
Le nombre de disques est entré en ligne 5Ø et refusé s'il y a plus de 7 disques.
La boucle I (7Ø-9Ø) permet de fabriquer une variable A$ (1) qui contient l'état de la première pile sous forme numérique (en ligne 3Ø, trois variables sont prévues pour conserver l'état des trois piles).
Les socles sont dessinés par la ligne 11Ø et les trois piles par les boucles emboîtées I (13Ø-18Ø) et J (14Ø-17Ø).
Le contrôle du coup est effectué par la ligne 27Ø et l'état des piles est modifié par les lignes 28Ø et 29Ø.

POUR JOUER :
Après un RUN qui lance le programme, on entre le nombre de disques demandé, suivi de ENTER. On obtient le dessin de l'état initial et, à la question TOUR DE DEPART, on répond en donnant le numéro de la tour de départ. Ce numéro (1, 2 ou 3) sera refusé si la tour est vide. Puis on tape ENTER et, à la question TOUR D'ARRIVEE, on répond par le numéro de la tour d'arrivée.
Si le coup n'est pas possible, on sera obligé de recommencer à TOUR DE DEPART. Si le coup est possible, le déplacement est réalisé et l'on recommence.

EXTENSIONS POSSIBLES :
On pourra dessiner des disques de différentes couleurs, introduire un compteur de coups et, surtout, définir un jeu pour l'ordinateur.
On peut aussi essayer de compliquer le jeu avec deux tiges de plus et deux joueurs, chacun ayant des disques de couleur différente, par exemple.

```
10 REM HANOI
20 CLS
30 DIM A(3):A(1)=9:A(2)=9:A(3)=9
40 D$="":D$=STRING$(9,233)
50 INPUT "NOMBRE DE DISQUES (MOINS DE 8)
:";N
60 IF N<2 OR N>7 THEN 50
70 FOR I=0 TO N-1
80 A(1)=VAL(STR$(A(1))+STR$(N-I))
90 NEXT I
100 CLS
110 LOCATE 1,15:PEN 5:PRINT D$:LOCATE 13
,15:PRINT D$:LOCATE 25,15:PRINT D$
120 LOCATE 5,15:PAPER 2:PRINT "1":LOCATE
 17,15:PRINT "2":LOCATE 29,15:PRINT "3":
PAPER 0:PEN 3
130 FOR I=0 TO N-1
140 FOR J=1 TO 3
150 B$=RIGHT$(STR$(A(J)),LEN(STR$(A(J)))
-1)
160 IF LEN(B$)>I+1 THEN LOCATE 12*J-10,1
4-I:PRINT LEFT$(D$,VAL(MID$(B$,I+2,1)))
170 NEXT J
180 NEXT I
190 PEN 1
200 LOCATE 1,18:PRINT CHR$(18);:INPUT "T
OUR DE DEPART :";D
210 IF D>3 OR D<1 THEN 200
220 IF A(D)=9 THEN 200
230 LOCATE 1,20:PRINT CHR$(18);:INPUT "T
OUR D'ARRIVEE :";A
240 IF A=D THEN 200 ELSE IF (A<1 OR A>3)
 THEN 230
250 B$=RIGHT$(STR$(A(D)),LEN(STR$(A(D)))
-1)
260 C$=RIGHT$(STR$(A(A)),LEN(STR$(A(A)))
-1)
270 IF RIGHT$(B$,1)>RIGHT$(C$,1) THEN 20
0
280 A(A)=VAL(C$+RIGHT$(B$,1))
290 A(D)=VAL(LEFT$(B$,LEN(B$)-1))
300 SOUND 1,478:SOUND 1,426:SOUND 1,379:
GOTO 100
```

Course de chevaux

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : aucune.

LE JEU :
On entrera un nombre de chevaux compris entre 1 et 9, et la course débutera immédiatement.
Il pourra y avoir des ex aequo, car l'on tiendra compte du fait que l'AMSTRAD s'occupe de faire avancer les chevaux les uns après les autres, en commençant par le numéro 1 ; le classement sera affiché au fur et à mesure des arrivées (et en tenant compte des ex aequo).

LE PROGRAMME :
Une première boucle non formelle (80-170) affiche les chevaux dans leur position de départ.
Une deuxième boucle I (100-180) les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alphanumérique A$.
La ligne 120 fera avancer les chevaux.
La ligne 140 gère les arrivées.
La ligne 150 réalisera les impressions.
Le sous-programme 240-280 fait avancer le compteur des arrivées, et imprime les résultats.

POUR JOUER :
On lance le programme par RUN.
A la question NOMBRE DE CHEVAUX : on répond par un entier de 1 à 9, suivi de ENTER.
La course se déroulera ensuite automatiquement.

EXTENSIONS POSSIBLES :
On pourra dessiner les chevaux avec un graphisme plus soigné, et les différencier par des numéros.
On pourra aussi imaginer tout un système de paris, qui interviendra avant le départ de chaque course.

```
10 REM COURSE DE CHEVAUX
20 RANDOMIZE(TIME)
30 INK 0,9:INK 2,0,26
40 CLS
50 INPUT "NOMBRE DE CHEVAUX :";N:IF N<0
OR N>9 THEN 50
60 A$="":P=0:A=0:T=1:K=0
70 J=T
80 A$=A$+"O":I=J
90 GOTO 150
100 FOR I=1 TO N
110 IF MID$(A$,I,1)="K" THEN 180
120 B$=CHR$(ASC(MID$(A$,I,1))+INT(RND*4+
1))
130 A$=LEFT$(A$,I-1)+B$+RIGHT$(A$,LEN(A$
)-I)
140 IF B$>="K" THEN GOSUB 240
150 LOCATE ASC(MID$(A$,I,1))-46,2*I+3:PR
INT "     ";:PAPER 3:PRINT I:PAPER 0
160 SOUND 1,159,1,4,0,0,1:FOR D=1 TO 60:
NEXT D
170 IF K=0 THEN J=J+1:IF J<=N THEN 80 EL
SE K=1:GOTO 100
180 NEXT I
190 P=P+T
200 T=0
210 IF A<N THEN 100
220 SOUND 1,478:SOUND 1,319:SOUND 1,379
230 LOCATE 1,24:END
240 A$=LEFT$(A$,I-1)+"K"+RIGHT$(A$,LEN(A
$)-I)
250 LOCATE 1,2*I+3:PEN 2:PRINT P:PEN 1
260 T=1:SOUND 1,239,2
270 A=A+T
280 RETURN
```

Accrochez les wagons

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : facile.

LE JEU :
Il faut essayer d'accrocher, le plus rapidement possible, la locomotive à son wagon, qui apparaissent tous deux en début de partie, en deux endroits plus ou moins éloignés de l'écran. On devra pour cela déplacer la locomotive en indiquant la suite des mouvements à effectuer (sous forme codée : H pour une case vers le haut ; B pour une case vers le bas ; D pour une case vers la droite et G pour une case vers la gauche).
On tapera, par exemple : BBGDHHH.

LE PROGRAMME :
Les lignes 4Ø à 7Ø dessinent la locomotive et le wagon.
La position initiale de la locomotive est définie par les lignes 9Ø et 1ØØ, celle du wagon par les lignes 11Ø et 12Ø.
La ligne 13Ø évite qu'ils soient initialement trop proches l'un de l'autre.
Les lignes 15Ø à 19Ø réalisent l'affichage du wagon et de la locomotive.
Le test de fin est en ligne 2ØØ.
L'utilisation de la fenêtre 1 permet de protéger l'écran contre un scrolling provoqué par la rentrée d'une trop longue chaîne R$ (lignes 25 et 21Ø).
Les mouvements à effectuer sont entrés dans une variable alphanumérique R$, en ligne 21Ø, et sont décodés dans une boucle I (24Ø-28Ø) pour être exécutés en ligne 27Ø.

POUR JOUER :
RUN permet d'afficher les positions de départ de la locomotive et du wagon.
On tape alors la série des déplacements à effectuer, sous la forme DDDHHHBHGG, suivie de ENTER.
La locomotive sera alors affichée dans sa nouvelle position, et l'on recommence, jusqu'à ce qu'elle soit accrochée au wagon.
On obtiendra alors l'affichage du score.

EXTENSIONS POSSIBLES :
Il sera intéressant de prévoir la perte de la partie si, par accident, au cours du déplacement, la locomotive vient en collision avec le wagon.
On pourra aussi prévoir plusieurs wagons et un ordre d'accrochage, jouer à plusieurs avec plusieurs locomotives, dessiner les voies ferrées, introduire des obstacles à contourner obligatoirement d'un certain côté, etc.

```
10 REM ACCROCHEZ LES WAGONS
20 RANDOMIZE(TIME)
25 WINDOW #1,1,40,24,25
30 DEFINT L,W
40 D$=CHR$(139)+CHR$(140)+CHR$(141)
50 A$=CHR$(142)+CHR$(143)+CHR$(143)
60 B$=CHR$(141)+CHR$(140)+CHR$(142)
70 C$="O O"
80 N=0
90 LY=1+RND*32
100 LX=2+RND*19
110 WY=1+RND*32
120 WX=2+RND*19
130 IF ABS(LX-WX)<4 THEN 90
140 CLS:PRINT CHR$(22)+CHR$(1)
150 PEN 2:LOCATE WY,WX:PRINT B$
160 LOCATE WY,WX+1:PRINT C$
170 PEN 3:LOCATE LY,LX-1:PRINT D$
180 LOCATE LY,LX:PRINT A$
190 LOCATE LY,LX+1:PRINT C$
200 IF LX=WX AND WY=LY-3 THEN 300
210 CLS #1:INPUT #1,"CHEMIN A SUIVRE (DG
GHBB....):";R$:R$=UPPER$(R$)
220 X=0:Y=0
230 N=N+1
240 FOR I=1 TO LEN(R$)
250 X=-(MID$(R$,I,1)="B" AND LX<22)+(MID
$(R$,I,1)="H" AND LX>2)
260 Y=-(MID$(R$,I,1)="D" AND LY<38)+(MID
$(R$,I,1)="G" AND LY>1)
270 LX=LX+X:LY=LY+Y
280 NEXT I
290 GOTO 140
300 CLS #1:PRINT #1,TAB(9) "BRAVO! TROUV
E EN";N;"COUPS"
310 SOUND 1,478:SOUND 1,319:SOUND 1,379
```

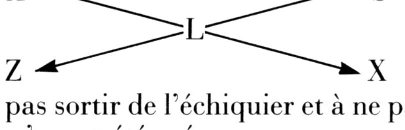
Le loup et les agneaux

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
C'est un jeu assez connu qui peut être pratiqué indifféremment sur un damier 8 sur 8, ou 1Ø sur 1Ø. Il est présenté, ici, dans une version pour un damier de 8 cases ; l'un des joueurs choisit le loup qui se déplace suivant les diagonales, en restant sur les cases noires, d'une case à chaque coup, et qui peut avancer ou reculer. Les agneaux se déplacent également sur les cases noires, en diagonale, mais n'ont pas le droit de reculer. Les cases déjà occupées sont interdites (pas de prises), et il n'y a pas de saut.
Le loup gagne s'il réussit à arriver sur la ligne du bas avant que tous les agneaux n'occupent celle du haut ; les agneaux gagnent dans ce dernier cas, ou s'ils arrivent à bloquer le loup.

LE PROGRAMME :
Une fenêtre de numéro 1, définie en ligne 2Ø, permet de protéger les saisies.
Une variable A$, en ligne 4Ø contient les positions de départ.
La boucle I (8Ø-17Ø) dessine le damier, et les lignes 19Ø à 22Ø placent le loup et les agneaux.
La ligne 23Ø gère le changement de joueur.
La ligne 25Ø affiche JEU DU LOUP si c'est le joueur qui manipule le loup qui a le trait, et JEU DES AGNEAUX dans le cas contraire.
Les lignes 28Ø à 3ØØ ou 31Ø à 32Ø analysent la réponse, en fonction du joueur concerné.

POUR JOUER :
RUN déclenche l'arrêt sur l'attente d'une entrée : Ø et ENTER si l'on veut que les agneaux commencent, et 1 pour le loup. Après quoi, le damier s'affiche et la phrase JEU DU LOUP (A, S, Z, X) pour le loup et JEU DES AGNEAUX (ND et NG) pour les agneaux.
Les agneaux répondent par un numéro, suivi de D ou G (pour droite ou gauche).
Ainsi 2D provoque le déplacement de l'agneau numéro 2 qui monte en diagonale d'une case sur la droite. Le déplacement effectué est affiché, puis on obtient une question identique pour le loup qui répondra en tapant A, S, Z ou X, suivant la direction choisie :



On devra veiller à ne pas sortir de l'échiquier et à ne pas venir sur l'adversaire, aucun contrôle n'ayant été prévu.

EXTENSIONS POSSIBLES :

On pourra prévoir un repérage des cases, un graphisme plus soigné, des contrôles de non dépassement des limites du terrain, de non venue sur une case déjà occupée...

On pourra aussi envisager de jouer sur un damier de 10 cases sur 10 cases, ce qui augmente légèrement la difficulté du jeu pour les agneaux. Mais l'extension la plus intéressante à réaliser sera certainement la réalisation d'un programme permettant à votre AMSTRAD de se comporter en adversaire honorable, soit avec le loup, soit avec les agneaux (et, dans ce cas, il sera possible de définir une stratégie gagnante pour AMSTRAD).

```
10 REM LE LOUP ET LES AGNEAUX
20 WINDOW #1,28,40,9,15
30 INPUT "AGNEAUX (0) OU LOUP (1)";S
40 A$="8183858714"
50 CLS
60 LOCATE 1,25:PRINT "    A  B  C  D  E
F  G  H";
70 FOR I=1 TO 8:LOCATE 1,3*I-1:PRINT I:N
EXT I
80 FOR I=1 TO 8
90 FOR J=1 TO 8
100 LOCATE 3*J,3*I-2
110 IF (I+J) MOD 2=0 THEN PAPER 2:ELSE P
APER 3
120 FOR N=1 TO 3
130 PRINT STRING$(3,233);CHR$(10);STRING
$(3,8);
140 NEXT N
150 PAPER 0
160 NEXT J
170 NEXT I
180 PEN 3
190 FOR I=1 TO 5
200 LOCATE 3*VAL(MID$(A$,2*I,1))+1,3*VAL
(MID$(A$,2*I-1,1))-1
210 IF I<>5 THEN PRINT RIGHT$(STR$(I),1)
;:ELSE PRINT "L";
220 NEXT I
230 PEN 1:CLS #1:S=ABS(S-1)
240 PRINT #1,"    JEU"
```

```
250 IF S=0 THEN PRINT #1,"  DU LOUP":PRI
NT #1," (A,S,Z,X)" ELSE PRINT #1,"DES  A
GNEAUX":PRINT #1,"  ND OU NG"
260 LOCATE #1,6,5:INPUT #1,"",R$:IF S=1
THEN GOTO 310
270 C=10
280 A=VAL(RIGHT$(A$,2))+(C+1 AND R$="X")
+(C-1 AND R$="Z")
290 A=A-(C+1 AND R$="A")-(C-1 AND R$="S"
)
300 GOTO 330
310 C=2*VAL(LEFT$(R$,1))
320 A=VAL(MID$(A$,C-1,2))-(9 AND RIGHT$(
R$,1)="D")-(11 AND RIGHT$(R$,1)="G")
330 MID$(A$,C-1,2)=RIGHT$(STR$(A),2)
340 GOTO 80
```

Niveau 4

Nom du programme	*Thèmes et particularités*
Tirage de dés	Suffisamment souple pour être utilisé partout.
La bombe	Saurez-vous la découvrir à temps ? (1 joueur).
Chapeaux	Soyez très observateur, et vous réussirez (2 joueurs).
Dessin de dés	Améliorer vos jeux de dés en incorporant ce programme.
Le singe	Un jeu de lettres pour deux joueurs, où il faut de bons réflexes.
Le jeu du 15	Un jeu de réflexion utilisant des chiffres.
Clavier	Apprenez à taper à la machine en vous amusant.
Adresse	Jeu de réflexe pour 1 joueur, très visuel.
Anagrammes	·Un bon jeu de réflexe : pour deux joueurs.
Zombies	Un peu farfelu, mais très visuel (1 joueur).
Ardoise magique	Mettez au point vos créations graphiques, sans papier ni crayon.
Décomaque	Pour vous éviter de sécher trop longtemps sur Nicomaque.
Evasion	Il vous faudra beaucoup de change pour réussir.
Répétitions	Retrouver une séquence de chiffres de plus en plus longue.
Mathieu	Vous ne savez pas dessiner ? Laissez faire votre ordinateur.
A travers	Un jeu de stratégie pour deux joueurs.
Calculatrice	Un programme de simulation très réaliste.
Chasse au sous-marin	Il vous faudra pas mal de réflexion, et beaucoup de chance.

Les techniques d'animation élémentaire

SAISIE AU VOL D'UN CARACTERE FRAPPE AU CLAVIER : INKE Y$

Le principe d'un jeu graphique, en animation, est de laisser le jeu se dérouler tout seul, d'une certaine manière, pendant un certain temps, tant qu'aucune intervention n'a été faite par le joueur.

Il faut donc disposer de la possibilité d'observer s'il y a eu ou non intervention du joueur, sans pour cela interrompre le déroulement du programme.

Dans le Basic AMSTRAD, cette propriété est offerte, par la **fonction INKEY$** qui effectue une scrutation des touches du clavier et enregistre le nom de la dernière touche frappée.

On pourra utiliser INKEY$ de deux façons :

- soit dans une boucle (réalisation d'un mouvement). On saisit INKEY$ et on le teste pour une éventuelle sortie de la boucle.
- soit dans une attente : tant que INKEY$ est vide (on n'a rien frappé) on attend, et l'on sort dès qu'une touche a été frappée.

Il sera quelquefois nécessaire de taper plusieurs fois la touche pour qu'elle soit prise en compte.

DEPLACEMENTS D'OBJETS DE BAS EN HAUT SUR L'ECRAN : SCROLL

Il est difficile, si l'on travaille en Basic, de gérer le déplacement simultané de plusieurs objets sur l'écran (sinon on obtient vite une animation saccadée, à cause d'un temps de déplacement prohibitif).

AMSTRAD réalise automatiquement un **"Scrolling"** (c'est-à-dire un déplacement de l'ensemble de l'image) vers le haut, lorsque l'on écrit sur la dernière ligne en bas de l'écran.

Cette propriété a été mise à profit dans plusieurs jeux : cela permet en effet, de déplacer d'une ligne vers le haut les divers objets (textes ou graphiques) présents sur l'écran à un moment donné. La ligne du haut est bien évidemment perdue.

Si un objet doit rester immobile sur l'écran, il sera nécessaire de l'effacer dans sa nouvelle position, et de le replacer à son emplacement initial.

Tirage de dés

REFERENCES :
Difficulté de programmation : aucune.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : aucune.

LE JEU :
Le programme proposé ne réalise pas un jeu en lui-même, mais pourra être utilisé en sous-programme pour de nombreux jeux.
Il offre simplement la possibilité de lancer de 1 à 9 dés, par simple pression d'une touche. Les dés sont ensuite affichés avec un graphisme très simplifié.

LE PROGRAMME :
La boucle WHILE/WEND (6Ø-8Ø) saisit le caractère frappé au clavier et la ligne 1ØØ le transforme en nombre (si ce n'est pas possible, le programme refuse l'entrée).
Une variable D non dimensionnée reçoit la valeur indiquée par chaque dé (ligne 15Ø).
Une boucle J (12Ø-14Ø) est utilisée pour dessiner un fond et une autre boucle I (11Ø-17Ø) sert au tirage (tirage en ligne 15Ø et impression en ligne 16Ø).
La ligne 19Ø offre la possibilité de refaire un tirage ; elle n'est pas indispensable.

POUR JOUER :
On lance le programme par l'ordre RUN et on obtient l'affichage NOMBRE DE DES A LANCER auquel on répond en frappant un chiffre.
Le résultat est alors automatiquement affiché ; il suffit de presser une touche quelconque pour recommencer le programme.
Pour arrêter, il est nécessaire de taper ESC deux fois.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, utiliser le canevas fourni par ce programme pour la réalisation de tout jeu se servant de dés (421, YAM'S, CRAPS, etc.).

```
10 REM TIRAGE DE DES
20 RANDOMIZE(TIME)
30 INK 0,9:INK 1,26:INK 2,0
40 CLS
50 LOCATE 1,1:PRINT "NOMBRE DE DES A LAN
CER :";:A$=""
60 WHILE A$="" OR A$<"1" OR A$>"9"
70 A$=INKEY$
80 WEND
90 PRINT CHR$(20);:PAPER 1:PEN 3:PRINT A
$:PAPER 3:PEN 2
100 N=VAL(A$)
110 FOR I=1 TO N
120 FOR J=1 TO 3
130 LOCATE 4*I-3,8+J:PRINT "   "
140 NEXT J
150 D(I)=1+INT(RND*6)
160 LOCATE 4*I-2,10:PRINT RIGHT$(STR$(D(
I)),1)
170 NEXT I
180 PAPER 0:PEN 1
190 GOTO 50
```

La bombe

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes censé trouver, à l'aide d'un détecteur spécialisé, un colis radioactif piégé disposé par un voyageur mal intentionné, dans une chambre d'hôtel. Seul problème, mais de taille : l'hôtel comporte un million de chambres (1ØØ*1ØØ*1ØØ).
Votre détecteur affiche, en permanence, un signal qui vous renseigne sur la proximité de la bombe, et le temps qui vous reste est affiché en permanence, décompté de 1Ø en 1Ø.
Pourrez-vous découvrir la bombe avant l'explosion ?

LE PROGRAMME :
Les fenêtres 1 et 2, définies en lignes 4Ø et 5Ø, serviront à l'affichage du signal et du temps.
Une variable numérique R, non dimensionnée, contient les trois coordonnées définissant la position du joueur.
La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, en ligne 7Ø.
La boucle I (9Ø-11Ø) dessine le détecteur.
Le signal de proximité est calculé en ligne 13Ø et la ligne 14Ø permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.
La ligne 15Ø permet l'affichage, à la fois du signal et du temps restant à courir.
La ligne 16Ø provoque de plus en plus fréquemment l'appel au sous-programme de son en 35Ø.
La ligne 17Ø réalise le décompte du temps.
La ligne 18Ø contrôle si l'on a trouvé ou non.
La boucle I (22Ø-26Ø) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 29Ø à 34Ø.

POUR JOUER :
Après le lancement du programme par RUN, le signal et le temps (en principe 2ØØ) sont affichés.
Le joueur est alors en position x=Ø, y=Ø, z=Ø, et on devra introduire successivement les trois coordonnées x, y, z de la chambre à explorer.
On fera, par exemple, 13 puis ENTER, 45 puis ENTER, etc.
Le nouveau signal est alors affiché, et le temps est décrémenté de 1Ø.
On continuera ainsi, soit jusqu'au bon résultat (arrêt du jeu et fin de programme), soit jusqu'à l'apparition de 3 nombres (x,y,z) sans sortie du programme, ce qui signifie alors que l'on a perdu.

EXTENSIONS POSSIBLES :

Il sera sans doute indispensable d'améliorer le graphisme, d'ajouter une explosion (sonore et visuelle) pour symboliser l'éclatement de la bombe, d'afficher symboliquement la position dans l'hôtel, etc.

```
10 REM LA BOMBE
20 RANDOMIZE(TIME)
30 DEF FN SS$(X)=RIGHT$(STR$(X),LEN(STR$
(X))-1)
40 WINDOW #1,13,19,7,7:WINDOW #2,22,24,7
,7
50 PAPER #1,1:PEN #1,3:PAPER #2,2:PEN #2
,0
60 L=100:T=200
70 A=INT(RND*L):B=INT(RND*L):C=INT(RND*L
)
80 CLS:PEN 3
90 FOR I=1 TO 3
100 LOCATE 11,5+I:PRINT STRING$(16,233)
110 NEXT I
120 PEN 1
130 S=ABS((A/L+B+C*L)-(R(1)/L+R(2)+R(3)*
L)):S=ROUND (S,2)
140 IF T=0 THEN PRINT "PERDU. LA SOLUTIO
N :";A;B;C:SOUND 1,253:SOUND 1,323:SOUND
 1,478,20:END
150 CLS #1:PRINT #1,FN SS$(S);:CLS #2:PR
INT #2,FN SS$(T);
160 EVERY INT(T/2) GOSUB 350
170 T=T-INT(SQR(L))
180 IF S=0 THEN LOCATE 1,21:PAPER 2:PEN
3:PRINT "BRAVO":SOUND 1,478:SOUND 1,319:
SOUND 1,379:END
190 LOCATE 1,14:PRINT " - X :"
200 LOCATE 1,16:PRINT " - Y :"
210 LOCATE 1,18:PRINT " - Z :"
220 FOR I=1 TO 3
230 R$=""
240 GOSUB 280
250 R(I)=VAL(R$)
260 NEXT I
270 GOTO 130
280 LOCATE 7,2*I+12:PRINT CHR$(243);CHR$
(18);
290 WHILE A$<>CHR$(13)
300 A$=INKEY$
310 PRINT A$;
320 R$=R$+A$
330 WEND
340 A$="":LOCATE 7,2*I+12:PRINT " ":RETU
RN
350 SOUND 2,159,1:RETURN
```

Chapeaux

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexe, pour deux joueurs.
Difficulté du jeu : facile.

LE JEU :
L'idée en est très simple : deux personnages se déplacent l'un vers l'autre sur l'écran et se croisent. Au moment où ils se croisent, il est possible qu'ils échangent leurs chapeaux. Il s'agit d'être le premier à dire si oui ou non il y a eu échange de chapeaux lors du croisement.

LE PROGRAMME :
La ligne 3Ø contient le dessin des deux chapeaux (dans deux variables alphanumériques C$(1) et C$(2)).
La ligne 4Ø contient les symboles qui permettront de dessiner les personnages sous forme de leur code ASCII à 3 chiffres.
La ligne 6Ø permet, s'il y a lieu, l'échange des chapeaux.
La variable T de la ligne 7Ø sert à repérer s'il y a eu échange ou non.
Le dessin des deux personnages est réalisé par une boucle L (9Ø-18Ø). La boucle imbriquée I réalise le dessin ligne par ligne et la boucle N assure le dessin de chaque ligne.
La ligne 19Ø permet l'affichage simultané des deux chapeaux.
Leur déplacement est réalisé par la ligne 2ØØ.
Le test, permettant ou non le changement des couvre-chefs, est effectué en ligne 22Ø.
Les lignes 25Ø à 31Ø déterminent si le premier joueur ayant appuyé sur une touche a gagné ou perdu.

POUR JOUER :
Lancer le programme par RUN.
Les personnages commencent immédiatement à se déplacer l'un vers l'autre.
Le joueur de gauche dispose des touches 1 et 2 et celui de droite, des touches 9 et Ø.
S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et Ø (à droite).
S'il n'y a pas eu de changement, il faudra, pour gagner, être le premier à appuyer sur 2 (à gauche) et sur 9 (à droite).
L'affichage indiquera alors BRAVO GAUCHE ou BRAVO DROITE, ou, encore, PERDU GAUCHE ou PERDU DROITE.

EXTENSIONS POSSIBLES :
Il sera possible de prévoir un affichage couleur, avec une animation plus complexe des personnages.
On pourra aussi compliquer le jeu à plaisir avec plusieurs objets échangés de façon aléatoire (il faudrait alors en trouver la liste complète).

```
10 REM CHAPEAUX
20 RANDOMIZE(TIME):PRINT CHR$(22);CHR$(1
)
30 C$(1)=CHR$(222)+CHR$(207)+CHR$(223):C
$(2)=CHR$(214)+CHR$(203)+CHR$(215)
40 A$= "032224032150159156149149149155514
9155150155156149032149149032149153032147
"
50 X(1)=1:X(2)=33
60 D$=C$(1):C$(1)=C$(2):C$(2)=D$
70 T=ABS(T-1)
80 CLS:INK 1,1
90 FOR L=1 TO 2
100 PEN 1:LOCATE X(L),2:PRINT C$(L)
110 PEN 1+L
120 FOR I=1 TO 8
130 LOCATE X(L),I+2
140 FOR N=0 TO 2
150 PRINT CHR$(VAL(MID$(A$,9*I-8+3*N,3))
);
160 NEXT N
170 NEXT I
180 NEXT L
190 INK 1,24
200 X(1)=X(1)+2:X(2)=X(2)-2:INK 1,24
210 TE=0:WHILE TE<200:TE=TE+1:WEND
220 IF X(1)=X(2) AND RND<0.5 THEN 60
230 IF X(2)=-1 THEN PRINT CHR$(7);:T=0:G
OTO 50
240 R$=INKEY$:IF R$="" THEN 80
250 PRINT CHR$(22);CHR$(0);:IF X(1)<=X(2
) THEN 280
260 PAPER 2:PEN 3:LOCATE 13,23:IF (R$="1
" AND T=0) OR (R$="2" AND T=1) THEN PRIN
T "BRAVO! GAUCHE":SOUND 1,478:SOUND 1,37
9:SOUND 1,319:PAPER 0:END
270 IF (R$="0" AND T=0) OR (R$="9" AND T
=1) THEN PRINT "BRAVO! DROIT":SOUND 1,47
8:SOUND 1,379:SOUND 1,319:PAPER 0:END
280 PAPER 3:PEN 1:LOCATE 14,23:IF R$="2"
 OR R$="1" THEN PRINT "PERDU GAUCHE"
290 IF R$="9" OR R$="0" THEN PRINT "PERD
U DROIT"
300 FOR I=1 TO 200 STEP 2:SOUND 1,I,1:NE
XT
310 PAPER 0
```

Dessin de dés

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
Le programme proposé est surtout intéressant par son côté graphique puisqu'il permet de représenter, de façon très convenable, de 1 à 5 dés sur l'écran, en utilisant une technique de dessin un peu particulière.
Chaque réponse à la question NOMBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

LE PROGRAMME :
La ligne 7Ø permet de contrôler s'il y a eu ou non action sur une touche de 1 à 5.
Les lignes 9Ø et 1ØØ initialisent les deux variables de dessin des dés.
Dans la boucle I (11Ø-2ØØ), les lignes 12Ø, 13Ø et 18Ø dessinent le cadre du dé.
La ligne 14Ø réalise le tirage, et la boucle J (15Ø-19Ø) dessine les points en relation avec le tirage réalisé (en appelant les variables A$ et B$).

POUR JOUER :
On lance le programme par l'ordre RUN, et, à la question NOMBRE DE DES, on répond en frappant une touche de 1 à 5. Les dés demandés sont alors représentés, et la question NOMBRE DE DES est de nouveau posée.
Il suffira de taper ESC deux fois pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra conserver la partie dessin utilisée dans ce programme pour réaliser toute sorte de jeux de dés, avec une représentation graphique de qualité convenable et facile à modifier si besoin était.

```
10 REM DESSIN DE DES
20 RANDOMIZE(TIME)
30 INK 0,9:INK 1,26:INK 2,15:INK 3,0
40 CLS
50 PEN 1:LOCATE 1,15:PRINT "NOMBRE DE DE
S :?";
60 A$=INKEY$:IF A$="" THEN 60
70 N=VAL(A$):IF N<1 OR N>5 THEN 40
80 CLS:PEN 3
90 A$=STRING$(7,32)+CHR$(231)+STRING$(2,
32)+CHR$(231)+STRING$(8,32)+CHR$(231)+CH
R$(231)+STRING$(3,32)+CHR$(231)
100 B$="121314324515525555"
110 FOR I=1 TO N
120 LOCATE 8*I-7,3:PRINT CHR$(150);STRIN
G$(5,154);CHR$(156)
130 LOCATE 8*I-7,9:PRINT CHR$(147);STRIN
G$(5,154);CHR$(153)
140 X=INT(RND*6+1)
150 FOR J=1 TO 5
160 IF J<=3 THEN S=5*VAL(MID$(B$,3*(X-1)
+J,1))-4
170 IF J<=3 THEN PEN 2:LOCATE 8*I-6,2*J+
2:PRINT MID$(A$,S,5):PEN 3
180 LOCATE 8*I-7,J+3:PRINT CHR$(149):LOC
ATE 8*I-1,J+3:PRINT CHR$(149)
190 NEXT J
200 NEXT I
210 GOTO 50
```

Le singe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
Il s'inspire de l'idée déjà ancienne du singe qui, frappant au hasard sur une machine à écrire, a des chances non nulles de finir par écrire un texte signifiant.
Le rôle du singe est, ici, tenu par AMSTRAD qui va afficher des groupes de 3 à 6 lettres pendant un temps relativement court.
Si l'un des joueurs estime qu'il peut réussir à faire un mot avec les lettres proposées, il doit taper rapidement sur sa touche de jeu (1 ou Ø).
Le mot devra alors être accepté ou refusé par l'adversaire qui répondra JUSTE ou FAUX.
Chaque mot juste rapporte 1 point, chaque mot faux enlève 1 point.

LE PROGRAMME :
La ligne 4Ø contient une liste V$ des voyelles courantes, et une liste C$ des consonnes courantes.
Il sera possible de modifier ces deux listes pour augmenter, par exemple, la fréquence d'apparition de certaines lettres (on pourra ainsi mettre deux E dans V$ ou deux R dans C$, etc.).
La ligne 5Ø définit la longueur du mot (en principe, de 3 à 6 lettres, mais on peut également modifier cette longueur).
La boucle I (8Ø-13Ø) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C$ (ligne 1ØØ), soit dans la liste V$ (ligne 11Ø).
La ligne 15Ø temporise l'apparition du résultat.
La ligne 17Ø gère la fin de partie, avec affichage du score.
La ligne 18Ø regarde si l'un des joueurs a réagi.
Enfin, les lignes 21Ø et 22Ø comptabilisent les points de chaque joueur.

POUR JOUER :
On lance le programme en tapant RUN.
La première série de lettres est alors affichée sur l'écran. Le joueur de gauche devra taper 1 s'il voit qu'il peut réaliser un mot, celui de droite devra taper Ø.
Si rien n'a été frappé, une autre série de lettres sera affichée après peu de temps.
Si l'un des joueurs a frappé Ø ou 1, la question (J)USTE ou (F)AUX est alors affichée en bleu pour le joueur de gauche et en rouge pour l'autre.
On répond J si le mot est correct, F si le mot est incorrect, suivi de ENTER.
Les points sont alors calculés par AMSTRAD, et la série de lettres suivante est alors affichée.
Pour sortir du programme, il faudra taper P au lieu de Ø ou 1 ; le score final est alors affiché.

EXTENSIONS POSSIBLES :
On pourra modifier la longueur des mots, la fréquence d'apparition des lettres, le décompte des points, la règle du jeu, etc.

```
10 REM LE SINGE
20 RANDOMIZE(TIME)
30 MODE 0
40 A=0:B=0:V$="AEIOU":C$="BCDGLMNPRSTV"
50 X=INT(RND*4+3)
60 M$=""
70 CLS
80 FOR I=1 TO X
90 E$=CHR$(INT(RND*26+65))
100 IF RND<0.4 THEN E$=MID$(C$,INT(LEN(C
$)*RND+1),1)
110 IF RND<0.5 THEN E$=MID$(V$,INT(LEN(C
$)*RND+1),1)
120 M$=M$+E$
130 NEXT I
140 LOCATE (20-LEN(M$))/2,6:PEN 6:PAPER
7:PRINT M$:SOUND 1,478:SOUND 1,426:SOUND
 1,379:PAPER 0:PEN 1
150 TE=0:WHILE TE<500:TE=TE+1:WEND
160 A$=INKEY$
170 IF A$="P" THEN MODE 1:LOCATE 4,12:PE
N 2:PRINT "SCORE DE A:";A;:PEN 3:PRINT "
- SCORE DE B:";B:SOUND 1,478:SOUND 1,379
:SOUND 1,319:TE=0:WHILE TE<1000:TE=TE+1:
WEND:END
180 IF A$<>"0" AND A$<>"1" THEN 50
190 LOCATE 2,18:PEN 3-VAL(A$):PRINT "(J)
USTE OU (F)AUX"
200 R$=INKEY$:IF R$="" THEN 200
210 A=A-(R$="J" AND A$="1")+(R$="F" AND
A$="1")
220 B=B-(R$="J" AND A$="0")+(R$="F" AND
A$="0")
230 GOTO 50
```

Le jeu du 15

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
C'est un jeu pour deux joueurs, assez ancien, qui était autrefois pratiqué dans les foires. On dispose d'un tapis de jeu de 9 cases, et chacun mise à son tour un jeton de sa couleur, sur une case disponible.
Le premier des deux joueurs qui, avec exactement 3 jetons, arrive à faire un total de 15, a gagné.

LE PROGRAMME :
La boucle I (5Ø-8Ø) dessine le terrain de jeu, et les lignes 9Ø à 1ØØ s'occupent de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton X ou O correspondant).
La ligne 11Ø attend le coup joué.
La boucle I (12Ø-14Ø) vérifie la validité du coup.
La ligne 15Ø teste la fin de partie, en cas de match nul, et la ligne 16Ø refuse les coups non conformes (avec retour à la situation initiale, ligne 9Ø).
La ligne 17Ø s'occupe de l'affichage du coup, et la ligne 18Ø l'enregistre dans une variable R$. La partie la plus intéressante du programme, qui pourrait servir à définir une stratégie machine, va de la ligne 2ØØ à la ligne 29Ø : on utilise 3 boucles emboîtées pour vérifier si l'un des joueurs a obtenu, avec 3 pions, un total de 15. Le changement de joueur s'effectue en ligne 3ØØ.

POUR JOUER :
Lancer le jeu en tapant RUN.
Le terrain de jeu s'affiche, puis la question JOUEUR 1 : X, et le joueur qui a choisi les X devra donner le numéro de la case où il désire jouer si elle est libre, son coup y sera enregistré.
On obtient, ensuite, l'affichage JOUEUR 2 : O et le deuxième joueur procède de la même façon. Si l'un des joueurs gagne (total de 15 sur 3 coups), il obtient l'affichage BRAVO et la partie s'arrête.
En cas de match nul, la machine affiche C'EST FINI.
On fera une autre partie en tapant RUN.

EXTENSIONS POSSIBLES :
Le plus intéressant sera sans doute d'essayer de définir une stratégie machine (en essayant d'éviter que la machine ne joue toujours le 5 lorsqu'elle doit commencer).
On peut aussi, bien sûr, améliorer le graphisme et introduire des effets sonores pour les différents messages.

```
10 REM LE JEU DU 15
20 CLS
30 F=0:R$="123456789"
40 O$="":X$=""
50 FOR I=0 TO 24 STEP 3
60 LOCATE I+6,1:PRINT CHR$(150)+CHR$(154
)+CHR$(156):LOCATE I+6,2:PRINT CHR$(149)
+" "+CHR$(149):LOCATE I+6,3:PRINT CHR$(1
47)+CHR$(154)+CHR$(153)
70 LOCATE I+6,4:PRINT (I+3)/3
80 NEXT I
90 LOCATE 1,10:PEN 2+F:PRINT "JOUEUR";F+
1;
100 PRINT ":"+CHR$(203+F*27)
110 A$=INKEY$:IF A$<"1" OR A$>"9" THEN 1
10
120 FOR I=1 TO LEN(R$)
130 IF A$=MID$(R$,I,1) THEN GOSUB 170
140 NEXT I
150 IF VAL(R$)=0 THEN LOCATE 15,20:PAPER
 3:PEN 1:PRINT "C'EST FINI":TE=0:WHILE T
E<200:SOUND 1,TE,1:TE=TE+1:WEND:END
160 GOTO 90
170 LOCATE 3*I+4,2:PRINT CHR$(203+F*27):
IF F=1 THEN O$=O$+A$ ELSE X$=X$+A$
180 MID$(R$,I,1)="0"
190 IF F=1 THEN T$=O$ ELSE T$=X$
200 IF LEN(T$)<3 THEN 300
210 FOR I=1 TO 3
220 IF I=LEN(T$) THEN 300
230 FOR J=I+1 TO LEN(T$)
240 IF J=LEN(T$) THEN 300
250 FOR K=J+1 TO LEN(T$)
260 IF VAL(MID$(T$,I,1))+VAL(MID$(T$,J,1
))+VAL(MID$(T$,K,1))=15 THEN 320
270 NEXT K
280 NEXT J
290 NEXT I
300 F=(F+1) MOD 2
310 RETURN
320 LOCATE 12,20:PAPER 2+F:PEN 0:PRINT "
BRAVO! JOUEUR ";CHR$(203+F*27)
330 SOUND 1,478:SOUND 1,478:SOUND 1,379:
SOUND 1,319
340 PEN 1:PAPER 0
```

Clavier

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexe (apprentissage du clavier).
Difficulté du jeu : difficile.

LE JEU :
Une reproduction du clavier de votre ordinateur apparaît sur l'écran ; une flèche apparaît en dessous d'un caractère, qu'il convient de frapper au clavier. Plus l'on frappe vite, et plus la flèche se déplace rapidement (le temps écoulé depuis le début s'affiche en fin de partie).

LE PROGRAMME :
Une variable A$ dimensionnée en ligne 5Ø, contiendra la liste des touches concernées. Les valeurs des touches sont entrées de la ligne 6Ø à la ligne 11Ø par appel d'un sous-programme situé en 3ØØ.
Les boucles emboîtées J (12Ø-16Ø) et I (13Ø-15Ø) dessinent le clavier.
La ligne 18Ø définit la position de la flèche imprimée par la ligne 19Ø.
La touche frappée est testée aux lignes 22Ø-23Ø et la flèche est effacée par la ligne 25Ø.
Les affichages de fin sont réalisés aux lignes 27Ø à 28Ø.

POUR JOUER :
On lance le programme par RUN. L'on dispose alors de quelques secondes pour frapper la touche qui est au-dessus de la flèche, puis une autre flèche est affichée, etc.
La durée de la partie n'est pas limitée ; il suffit de taper Ø pour obtenir l'affichage du score, et du temps total.
Faire RUN pour une autre partie.

EXTENSIONS POSSIBLES :
On pourra envisager de faire clignoter les lettres, ou de les afficher en couleur, au lieu d'indiquer la lettre par une étoile. On pourra afficher les lettres devinées justes en bas de l'écran, et éventuellement s'entraîner à taper un texte qui défile en haut de l'écran, lettre par lettre.

```
10 REM CLAVIER
20 RANDOMIZE(TIME)
30 MODE 0
40 T=0:S=0:P=0
50 DIM A$(10,3)
60 S$="QWERTYUI":J=1
70 GOSUB 300
80 S$="ASDFGHJK":J=2
90 GOSUB 300
100 S$="ZXCVBNM,":J=3
110 GOSUB 300
120 FOR J=1 TO 3
130 FOR I=1 TO 8
140 PEN I: LOCATE 2*I+J,5*J:PRINT A$(I,J
)
150 NEXT I
160 NEXT J
170 T=TIME
180 X=INT(RND*8+1):Y=INT(RND*3+1)
190 LOCATE 2*X+Y,5*Y+2:PEN X:PRINT CHR$(
240)
200 FOR F=1 TO 50+RND*80:A$=INKEY$
210 IF A$="" THEN NEXT F
220 IF A$=A$(X,Y) THEN S=S+1
230 IF A$="O" THEN 270
240 P=P+1
250 LOCATE 2*X+Y,2+5*Y:PRINT " "
260 GOTO 180
270  MODE 1:PEN 3:LOCATE 3,12:PRINT USIN
G "### TROUVES JUSTES SUR ### ESSAIS";S,
P
280 LOCATE 11,14:PRINT USING "VOTRE TEMP
S : ####";ROUND((TIME-T)/300)
290 SOUND 1,478:SOUND 1,379:SOUND 1,319:
END
300 FOR I=1 TO 8:A$(I,J)=MID$(S$,I,1):NE
XT I:RETURN
```

Adresse

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

LE JEU :
Des chiffres, entre 0 et 9, vont apparaître au hasard, en divers emplacements de l'écran. Ils restent affichés pendant un temps très court. Le joueur doit réussir à appuyer sur la touche numérique correspondante, au moment où le chiffre est affiché.
Le chiffre tapé par le joueur sera alors affiché en grand format sur la gauche de l'écran ; et chaque réponse exacte rapporte 10 points. La durée de la partie est aléatoire et le score sera affiché en fin de partie.

LE PROGRAMME :
Les lignes 50 et 60 contiennent, dans deux variables alphanumériques C$ et B$, les références qui seront utilisées pour dessiner les chiffres en grand format.
L'emplacement du chiffre tiré par AMSTRAD est choisi aux lignes 80 et 90, et le chiffre est tiré en ligne 110.
La ligne 120 réalise l'impression, et la ligne 130 détermine la durée d'affichage. La boucle WHILE/WEND (70-170) définit la durée du jeu, et la fin de partie est traitée par la ligne 190.
La boucle I (220-270) réalise le dessin grand format du chiffre frappé par le joueur et l'imprime dans la fenêtre ≠ 2 définie en ligne 30.
Le contrôle de l'exactitude du résultat est effectué par la ligne 280.

POUR JOUER :
Le programme sera lancé par RUN, puis un premier chiffre apparaît, après une courte pause, vers le milieu de l'écran. Le joueur devra réagir suffisamment vite, tant qu'il reste à l'affichage, et le retaper. Le chiffre qu'il a tapé se dessinera alors en haut à gauche de l'écran, en grand format.
Si le résultat est exact, le joueur marquera 10 points.
Le temps de déroulement de la partie sera fonction des chiffres qui apparaîtront à l'affichage.
En fin de partie, le score sera affiché. Il sera possible de s'arrêter à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.
On pourra aussi enlever des points à chaque résultat faux, modifier les règles de calcul du score et la durée du jeu, etc.

```
10 REM ADRESSE
20 RANDOMIZE(TIME)
30 WINDOW #2,2,10,7,18:PEN 3:PAPER #2,1:
PEN #2,0
40 T=100:S=0:C$=CHR$(143):CLS
50 C$=C$+C$+C$+C$+STRING$(4,32)+C$+C$+"
"+C$
60 B$="14441333331312113131441331213112l
411333314141141 31"
70 WHILE T>=0
80 X=12+INT(RND*20)
90 Y=1+INT(RND*23)
100 C=2+INT(RND*25)
110 N=INT(RND*10)
120 LOCATE X,Y:INK 1,C:PRINT N;CHR$(7)
130 FOR I=1 TO RND*350+400:NEXT I
140 A$=INKEY$:IF A$<>"" THEN 210
150 T=T-N
160 LOCATE X,Y:PRINT "   ";
170 WEND
180 CLS
190 PRINT:PRINT "VOTRE SCORE EST DE";S;"
SUR 100"
200 SOUND 1,478:SOUND 1,426:SOUND 1,379:
END
210 U=VAL(A$):CLS #2
220 FOR I=1 TO 5
230 A=3*VAL(MID$(B$,5*U+I,1))-2
240 FOR K=1 TO 2
250 LOCATE #2,4,2*I+K-1:PRINT #2,MID$(C$
,A,3)
260 NEXT K
270 NEXT I
280 IF U=N THEN S=S+10
290 GOTO 150
```

Anagrammes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (2 joueurs).
Difficulté du jeu : selon le niveau.

LE JEU :
Des lettres sont choisies au hasard par AMSTRAD, le nombre de lettres étant fonction du niveau de difficulté désiré. Puis les lettres du mot ainsi obtenu sont brouillées, et affichées au bas de l'écran ; un nouveau mot est alors confectionné, de façon aléatoire, à partir des mêmes lettres et il est affiché sous le premier, et ainsi de suite ; les deux joueurs (gauche et droit) disposent chacun d'une touche (1 ou Ø) et doivent la presser dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat sera délivré.

LE PROGRAMME :
Le niveau du jeu est entré à la ligne 5Ø. La boucle I (8Ø-1ØØ) réalise le choix des lettres.
Une deuxième boucle I (12Ø-19Ø) effectue le brouillage, et la ligne 14Ø permet de détecter la pression d'une touche.
La ligne 2ØØ affiche un nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.
La ligne 24Ø cherche quel est celui des deux joueurs qui a pressé une touche.
La boucle I (25Ø-27Ø) explore les mots présents sur l'écran ligne par ligne. On obtient ainsi un mot qui est comparé au mot initial en ligne 26Ø et le résultat BRAVO apparaît si c'est exact.
Sinon, la partie s'arrête sur l'affichage PERDU (ligne 28Ø).

POUR JOUER :
On lance le programme par RUN.
On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire ENTER. Les mots s'affichent en partant du bas de l'écran, et montent au fur et à mesure ; si l'un des joueurs croit reconnaître dans le mot qui apparaît un des mots déjà affichés, il devra immédiatement presser sa touche (1 ou Ø), et son résultat sera affiché après un court instant.
La partie sera alors terminée, on fera ESC deux fois pour récupérer le contrôle du clavier.

EXTENSIONS POSSIBLES :
On pourra utiliser d'autres affichages que des mots, modifier les temps d'affichage, faire participer plus de deux joueurs, améliorer le son et la couleur, etc.

```
10 REM ANAGRAMMES
20 RANDOMIZE(TIME)
30 DIM A$(24)
40 CLS
50 INPUT "NIVEAU (DE 1 A 3) :";N
60 IF N<1 OR N>3 THEN 40
70 S$=""
80 FOR I=1 TO N+3
90 S$=S$+CHR$(INT(RND*26+65))
100 NEXT I
110 A$="":B$=S$
120 FOR I=1 TO LEN(S$)
130 N=INT(RND*LEN(S$)+1)
140 D$=INKEY$
150 IF D$="1" OR D$="0" THEN 240
160 IF MID$(B$,N,1)="-" THEN 130
170 A$=A$+MID$(B$,N,1)
180 MID$(B$,N,1)="-"
190 NEXT I
200 LOCATE 1,25:PRINT A$:PRINT CHR$(8)
210 FOR I= 1 TO 23:A$(I)=A$(I+1):NEXT I:
A$(24)=A$
220 S$=A$
230 GOTO 110
240 IF D$="1" THEN H$="JOUEUR DE GAUCHE"
 ELSE H$="JOUEUR DE DROITE"
250 FOR I=1 TO 23
260 IF A$(I)=S$ THEN LOCATE LEN(S$)+4,I:
PEN 2:PRINT "BRAVO! ";H$:SOUND 1,478:SOU
ND 1,379:SOUND 1,319:GOTO 290
270 NEXT I
280 PEN 3:LOCATE LEN(S$)+4,24:PRINT "PER
DU....";H$:FOR I=1 TO 6:SOUND 1,400,10,6
*(I MOD 2):NEXT I
290 GOTO 290
```

Zombies

REFERENCES :
Difficulté de programmation : assez difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez facile.

LE JEU :
Les ZOMBIES sont des êtres malfaisants qui vont essayer de vous détruire, mais vous allez, vous aussi, essayer de les utiliser à votre profit, pour débarrasser votre espace vital des trous noirs qui l'obscurcissent. Sur l'écran, est tracé un cadre qui définit votre espace vital. Dans ce cadre, vont apparaître de façon aléatoire, les ZOMBIES (ce sont des Z), les trous noirs (des carrés bleus remplis de jaune) et le joueur (une tête). Les ZOMBIES convergent vers le joueur pour essayer de le capturer, mais celui-ci se déplace plus rapidement qu'eux, et il va essayer de leur faire avaler tous les trous noirs, sans se faire prendre.
La partie se termine, soit avec la disparition du joueur, soit par l'élimination de tous les ZOMBIES.

LE PROGRAMME :
La position des ZOMBIES est définie par un tableau Z non dimensionné (c'est inutile en dessous de 1Ø).
Les lignes 5Ø à 12Ø (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.
La ligne 13Ø définit la position initiale du joueur.
Une deuxième boucle I (15Ø-24Ø) réalise le déplacement du joueur.
Les lignes 29Ø à 34Ø contrôlent le déplacement du joueur.

POUR JOUER :
Après avoir lancé l'exécution par RUN, le joueur devra se déplacer en utilisant les 4 flèches et en veillant à ne pas sortir des limites du terrain.
Il devra manœuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coincer.
Pour arrêter la partie avant la fin, il suffira de taper ESC deux fois.

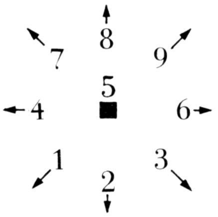
EXTENSIONS POSSIBLES :
Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles, de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle.
On pourra aussi, bien sûr, améliorer la qualité du dessin et introduire la couleur.

```
10 REM ZOMBIES
20 RANDOMIZE(TIME)
30 BORDER 9:INK 0,9:INK 3,1
40 CLS
50 FOR I=1 TO 22
60 LOCATE I,1:PRINT CHR$(233)
70 LOCATE 1,I:PRINT CHR$(233):LOCATE 23,
I:PRINT CHR$(233)
80 IF RND<0.5 THEN LOCATE I,INT(RND*20+2
):PEN 1:PAPER 3:PRINT CHR$(233);:PAPER 0
:PEN 1
90 S=INT(RND*14+2)
100 Z(INT(I/3)+1)=5+S
110 LOCATE I+1,22:PRINT CHR$(233);
120 NEXT I
130 K=INT(RND*18+2)
140 P=0
150 FOR I=1 TO 4
160 P=P+Z(I)
170 IF Z(I)=0 THEN 240
180 LOCATE Z(I+4),Z(I):PRINT " ";
190 Z(I)=Z(I)-SGN(Z(I)-S)
200 Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
210 IF TEST((Z(I+4)-1)*16+8,(25-Z(I))*16
+8)=1 THEN Z(I)=0:GOTO 230
220 LOCATE Z(I+4),Z(I):PEN 3:PRINT "Z"
230 IF Z(I)=S AND Z(I+4)=K THEN LOCATE 2
5,12:PEN 3:PRINT "VOUS AVEZ PERDU":WHILE
 TE<200:SOUND 1,TE,1:TE=TE+1:WEND:END
240 NEXT I
250 IF P=0 THEN LOCATE 2,23:PRINT "BRAVO
! VOUS AVEZ TUE TOUS LES ZOMBIES.":SOUND
 1,478:SOUND 1,426:SOUND 1,379:SOUND 1,4
78:SOUND 1,426:END
260 LOCATE K,S:PEN 2:PRINT CHR$(224)
270 FOR I=1 TO 1200:NEXT I
280 LOCATE K,S:PRINT " "
290 A$=INKEY$
300 K=K-2*((A$=CHR$(&F3))-(A$=CHR$(&F2))
)
310 IF K<=1 OR K>=23 THEN LOCATE 25,12:P
EN 3:PRINT "DOMMAGE!";CHR$(7);:END
320 S=S-2*((A$=CHR$(&F1))-(A$=CHR$(&F0))
)
330 IF S<=1 OR S>=22 THEN K=25:GOTO 310
340 GOTO 140
```

Ardoise magique

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : dessin.
Difficulté du jeu : facile.

LE JEU :
Comme le titre peut sembler l'indiquer, il s'agit, tout simplement d'utiliser l'écran du téléviseur comme une ardoise, en manipulant un point sur l'écran suivant 8 directions, à l'aide des touches de 1 à 9 du pavé numérique, en utilisant la convention suivante :



On peut choisir la couleur du point en appuyant sur une touche de A à N ; et il est possible d'effacer tout l'écran en appuyant sur ENTER.
On dispose ainsi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche A.

LE PROGRAMME :
La ligne 30 transforme les touches du clavier numérique en touches à répétition automatique de la frappe.
La ligne 40 contient le choix des possibilités de couleur sous forme hexadécimale.
La ligne 60 imprime le point élémentaire clignotant.
Les lignes 80 à 90 contrôlent les entrées.
Les lignes 100 à 130 provoquent le déplacement.
La ligne 140 vérifie que l'on ne sort pas des limites de l'écran.
La ligne 160 imprime le point.
Le sous-programme 180-190 choisit la couleur.

POUR JOUER :
On lance le programme par RUN ; un point apparaît vers le milieu de l'écran.
Si on frappe une lettre de A à N, la couleur du point change.
La touche ENTER efface tout, et l'on recommence.
On peut sortir à tout moment, en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, envisager d'écrire un programme identique en utilisant des tracés de segments, prévoir un sous-programme qui permette la sauvegarde du dessin réalisé pour une utilisation ultérieure, etc.

```
10 REM ARDOISE MAGIQUE
20 MODE 0:INK 15,26,0
30 FOR I=1 TO 20: KEY DEF I,1:NEXT I
40 C$="0123456789ABCD"
50 CLS:ORIGIN 320,200:A=0:B=0:C=4:X=0:Y=
0
60 PLOT X,Y,15
70 A$=INKEY$:IF A$="" THEN 70
80 IF  A$>="A" AND A$<="N" THEN GOSUB 18
0:GOTO 60
90 IF A$=CHR$(13) THEN GOTO 50
100 A=VAL(A$):U=(A-1) MOD 3:V=INT((A-1)/
3) MOD 3
110 U=U-1:V=V-1
120 A=X:B=Y
130 X=X+U:Y=Y+V
140 D=-(ABS(X)<319 AND ABS(Y)<199)
150 IF D=0 THEN X=A:Y=B
160 PLOT A,B,C
170 GOTO 60
180 C=VAL(CHR$(38)+MID$(C$,ASC(A$)-64,1)
)
190 RETURN
```

Décomaque

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solution du programme Nicomaque.
Difficulté du jeu : aucune.

LE JEU :
C'est une réplique au programme Nicomaque, proposé dans les pages précédentes ; vous entrez les restes successivement dans les divisions par 3, 5 et 7 d'un entier compris entre Ø et 1ØØ, et AMSTRAD calcule et affiche cet entier. Le procédé retenu peut d'ailleurs être généralisé (à vous de trouver comment).

LE PROGRAMME :
Les lignes 3Ø à 5Ø affichent quelques explications.
La ligne 1ØØ attend que vous frappiez un reste, qui sera refusé s'il est incompatible avec la division par le nombre affiché.
La ligne 11Ø emmagasine les restes dans une variable non dimensionnée A (S).
La ligne 14Ø réaffiche les restes entrés.
La boucle I (15Ø-17Ø) dessine des petits carrés selon les restes introduits.
Les lignes 21Ø à 22Ø gèrent la fin de partie.

POUR JOUER :
On lance le programme par RUN. Un texte de présentation est affiché, puis le message "RESTE DES DIVISIONS : 3, 5, 7" s'affiche.
On doit répondre Ø, 1 ou 2 pour 3, puis de Ø à 4 pour 5 et de Ø à 6 pour 7.
La solution est affichée dès que l'on a donné les 3 restes.
Pour une autre partie, taper O à la question VOULEZ-VOUS UN AUTRE ESSAI (O/N).
Tapez N pour finir.

EXTENSIONS POSSIBLES :
On peut étendre le programme aux entiers de 1 à 1ØØØ, en utilisant les restes par 3, 5, 7 et 11 ; on pourra rajouter du son et de la couleur pour rendre le programme plus attrayant.

```
10 REM DECOMAQUE
20 CLS:INK 2,9:PEN 2
30 PRINT "VOUS DEVEZ  ENTRER LES TROIS R
ESTES DANS"
40 PRINT "LES DIVISIONS PAR 3, 5 ET 7, D
'UN NOMBRE"
50 PRINT "INFERIEUR A 100."
60 I=3:S=1
70 WHILE I<8
80 LOCATE 5,10:SOUND 1,478:SOUND 1,426:S
OUND 1,379
90 PEN (I-3)/2:PAPER 3:PRINT "RESTE DANS
 LA DIVISION PAR";I;
100 A$=INKEY$:IF A$="" OR A$<"0" OR A$>C
HR$(47+I) THEN GOTO 100
110 A(S)=ASC(A$)-48
120 S=S+1:I=I+2
130 WEND
140 LOCATE 1,16:PAPER 0:PEN 1:PRINT "RES
TE :       PAR 3     PAR 5     PAR 7"
150 FOR I=1 TO 3
160 LOCATE 5+9*I,18:PAPER 3:PEN (I-1):PR
INT STRING$(A(I),233)
170 NEXT I
180 T=A(1)*70+A(2)*21+A(3)*15
190 IF T>105 THEN T=T-105:GOTO 190
200 LOCATE 10,22:PAPER 1:PEN 2:PRINT " L
A SOLUTION EST";T
210 PAPER 0:PEN 1:LOCATE 4,24:INPUT "VOU
LEZ-VOUS UN AUTRE ESSAI (O/N)";R$
220 IF UPPER$(R$)="O"THEN RUN
```

Evasion

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : facile.

LE JEU :
Votre seul problème : réussir à traverser l'écran, sans qu'AMSTRAD vous ait auparavant touché par l'un des nombreux obus explosifs qu'il va tirer sur vous.
Vous pouvez vous déplacer dans 4 directions (à l'aide des 4 flèches), et vous devez réussir à sortir du terrain par le bord droit, sans, bien sûr, avoir été touché une seule fois.

LE PROGRAMME :
La ligne 6Ø définit la position de départ du joueur.
La ligne 8Ø choisit la coordonnée horizontale de la case où AMSTRAD va tirer (liée à votre position), et la ligne 9Ø choisit la coordonnée verticale (arbitraire) de cette case.
Le déplacement du joueur est contrôlé par les lignes 17Ø et 18Ø . Sa position actuelle (sur l'écran, un symbole graphique) est imprimée par la ligne 19Ø.
La ligne 2ØØ regarde s'il est touché ou non.
La boucle WHILE/WEND (7Ø-21Ø) effectue le contrôle de fin de partie.

POUR JOUER :
On commence l'exécution du programme par RUN.
Un premier tir a lieu, et ce n'est qu'après que le joueur apparaît, représenté par un symbole graphique.
Les 4 touches de direction (←, ↓ , ↑ ,→) servent à déplacer le joueur.
S'il est touché, on obtient, sur l'écran, le message PERDU et la partie s'arrête.
S'il réussit à sortir par la droite de l'écran sans être touché, on obtient le message BRAVO, et la partie est également terminée.
On peut terminer à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
La couleur et un graphisme plus soigné amélioreront considérablement l'attrait du jeu. On pourra aussi rendre le tir des bombes plus aléatoire, introduire des obstacles à contourner, ajouter des contrôles, faire participer plusieurs joueurs, etc.

```
10 REM EVASION
20 RANDOMIZE(TIME)
30 DEFINT H,K,X,Y
40 CLS
50 ORIGIN 0,200
60 X=1:Y=13
70 WHILE X<38
80 H=16*(X-2)+32*RND
90 K=184-RND*368
100 IF X<4 THEN 150
110 FOR J=1 TO 3+RND*12
120 PLOT H+16*RND,K+16*RND,3
130 SOUND 1,834,1,7,0,0,1
140 NEXT J
150 LOCATE X,Y:PRINT "."
160 A$=INKEY$
170 X=X-(A$=CHR$(&F3))+(A$=CHR$(&F2))
180 Y=Y-(A$=CHR$(&F1))+(A$=CHR$(&F0))
190 LOCATE X,Y:PEN 2:PRINT CHR$(249):PEN
 1
200 IF X>3 AND (ABS(H-16*(X-1))+ABS(K-16
*(12-Y))<=15) THEN PEN 3:PRINT "PERDU!":
WHILE S<254:SOUND 1,S,1:S=S+1:WEND:END
210 WEND
220 LOCATE 1,23:PRINT "BRAVO!":SOUND 1,4
78:SOUND 1,379:SOUND 1,319
```

Répétitions

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de mémoire (1 ou plusieurs joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Des nombres vont apparaître un par un.
Le joueur devra à chaque coup les répéter tous, depuis le début (ce qui signifie qu'il en aura un de plus à taper à chaque fois).
Il devra essayer de tenir le plus longtemps possible. Le meilleur score est mémorisé après chaque partie, ce qui permettra éventuellement de jouer à plusieurs.

LE PROGRAMME :
La ligne 40 appelle un sous-programme (lignes 1000 à 1060) qui donne la règle du jeu.
La ligne 70 fabrique les chiffres, qui sont ensuite stockés dans une variable alphanumérique A$.
La ligne 80 permet l'impression du dernier chiffre au hasard sur la quatrième ligne de l'écran, et la durée de l'affichage est déterminée par la ligne 90.
La ligne 110 attend la réponse du joueur, qui est comparée à la réponse correcte en ligne 120.
La ligne 150 marque les points.
La ligne 160 contrôle le meilleur score.
Les affichages de fin de partie sont effectués de la ligne 200 à la ligne 230.

POUR JOUER :
Le lancement du programme par RUN provoque l'affichage de la règle du jeu, puis d'un chiffre qui disparaît. L'appareil pose alors la question VOTRE SEQUENCE.
On lui répond en redonnant le chiffre, suivi de ENTER. Si c'est exact, on a un court instant l'affichage de EXACT. ATTENTION, puis un nouveau chiffre est affiché, qui s'efface, et de nouveau la question VOTRE SEQUENCE. On retape alors les deux chiffres déjà obtenus, suivis de ENTER.
On continue ainsi (3 chiffres, puis 4, 5, etc.) jusqu'à ce que l'on fasse une erreur, ce qui provoque l'affichage du score (et du meilleur score obtenu jusqu'à présent), suivi de la question AUTRE PARTIE (O/N) ; on tape O pour continuer.
On peut aussi quitter le programme à tout moment, en tapant ESC deux fois avant l'affichage d'un chiffre.

EXTENSIONS POSSIBLES :
On pourra réaliser des jeux analogues et plus ou moins perfectionnés en remplaçant les chiffres par des lettres ou des couleurs ou des sons...

```
10 REM REPETITIONS
20 RANDOMIZE(TIME)
30 M=0
40 GOSUB 1000
50 A$=""
60 MODE 0:H=0
70 X=INT(RND*10):INK 3,2+INT(25*RND):A$=
A$+RIGHT$(STR$(X),1)
80 CLS:LOCATE 1+RND*20,4:PEN 3:PRINT X:P
EN 1
90 FOR I=1 TO 200+RND*300:NEXT I
100 CLS
110 LOCATE 4,8:PEN 6:PRINT "VOTRE SEQUEN
CE":PRINT:PEN 4:INPUT "";R$
120 IF R$<>A$ THEN 180
130 PRINT:SOUND 1,478:SOUND 1,426:SOUND
1,379:PRINT "  EXACT. ATTENTION"
140 FOR I=1 TO 400:NEXT I
150 H=H+1
160 IF M<H THEN M=H
170 GOTO 70
180 MODE 1:INK 0,6:BORDER 6:PEN 2
190 TE=0:WHILE TE<200:SOUND 1,TE,1:TE=TE
+1:WEND
200 INK 0,1:BORDER 1:PRINT "VOTRE SCORE
EST DE   ";H
210 PRINT:PRINT "MEILLEUR SCORE ACTUEL";
M
220 PRINT:INPUT "AUTRE PARTIE (O/N)";R$
230 IF R$="N" THEN PRINT CHR$(7);:END
240 GOTO 50
1000 CLS:PEN 1
1010 PRINT "VOUS DEVEZ RESPECTER TOUTE L
A SERIE DES"
1020 PRINT:PRINT "NOMBRES APPARUS DEPUIS
 LE  DEBUT DU JEU"
1030 PEN 2:LOCATE 4,12:PRINT "IL Y EN A
UN DE PLUS A CHAQUE FOIS"
1040 LOCATE 4,25:PAPER 1:PEN 0:PRINT "AP
PUYER SUR <ENTER> POUR COMMENCER";:PAPER
 0
1050 INPUT "",R$
1060 RETURN
```

Mathieu

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin aléatoire.
Difficulté du jeu : aucune.

LE JEU :
On utilise la définition d'une courbe en coordonnées polaires et la fonction DRAW pour obtenir sur l'écran des dessins aléatoires (de type fleur), plus ou moins élégants.
Les deux paramètres principaux A et B pourront être, si nécessaire, entrés à la main (instructions INPUT) pour faire une étude plus systématique des différentes courbes possibles.

LE PROGRAMME :
Les 3 premiers paramètres Z, K et T, introduits en ligne 5Ø, définissent le rayon maximum et l'occupation de l'écran.
La boucle WHILE/WEND (7Ø-9Ø) permet de choisir les 4 couleurs de fond d'encre, validées en ligne 1ØØ.
Les variables aléatoires A et B, choisies entre Ø et 2, de la ligne 11Ø, définissent l'allure de la courbe ; il pourra être intéressant de les introduire manuellement (par exemple : 11Ø INPUT A, B).
La boucle H (12Ø-17Ø) réalise le tracé, point par point. Le rayon R est calculé en ligne 13Ø, et le passage en coordonnées cartésiennes est effectué aux lignes 14Ø et 15Ø.
L'affichage des points est réalisé en ligne 16Ø.
La ligne 18Ø imprime un message de fin.
La ligne 21Ø permet de refaire un nouveau dessin, alors que les lignes 22Ø et 23Ø assurent un nouveau tracé sans effacement du précédent.

POUR JOUER :
On lance le programme en tapant RUN.
Le dessin sera ensuite exécuté automatiquement.
Le message "c'est fini" signale la fin du tracé.
Si on veut un nouveau tracé, on tape Ø.
Si on veut un tracé sans effacement du dessin précédent, on tape une touche quelconque.
Pour arrêter, faire ESC deux fois.

EXTENSIONS POSSIBLES :
On pourra essayer d'utiliser d'autres fonctions pour réaliser des tracés encore plus extraordinaires (par exemple avec simulation de la troisième dimension).

```
10 REM MATHIEU
20 RANDOMIZE(TIME)
30 ORIGIN 320,200
40 DEG
50 Z=7:K=13:T=3:CLS
60 F=0:E(1)=0:E(2)=0:E(3)=0
70 WHILE (E(1)-F)*(E(2)-F)*(E(3)-F)*(E(1
)-E(2))*(E(1)-E(3))*(E(2)-E(3))=0
80 F=INT(RND*27):E(1)=INT(RND*27):E(2)=I
NT(RND*27):E(3)=INT(RND*27)
90 WEND
100 INK 0,F:INK 1,E(1):INK 2,E(2):INK 3,
E(3)
110 A=3*RND:B=3*RND
120 FOR H=0 TO 360
130 R=Z*SIN(H*T)
140 X=INT(K*R*COS(A*H))
150 Y=INT(K*R*SIN(B*H))
160 DRAW X,Y,C+1
170 NEXT H
180 LOCATE 1,21:PRINT "C'EST FINI...."
190 SOUND 1,478:SOUND 1,379:SOUND 1,319
200 A$=INKEY$:IF A$="" THEN 200
210 IF UPPER$(A$)="O" THEN 50
220 C=(C+1) MOD 3
230 GOTO 110
```

A travers

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Sur l'écran apparaît un carré de 9 cases, avec sur le bord vertical des lettres et sur le bord horizontal des chiffres.
L'un des joueurs manipule les chiffres, l'autre les lettres. 3 déplacements sont possibles pour les chiffres, et 3 autres pour les lettres. Les lettres peuvent se déplacer vers le haut, à droite et à gauche et les chiffres vers la droite, en haut ou en bas.
Le gagnant est le premier qui arrive à faire sortir tous ses symboles (les lettres sortent en haut, et les chiffres à droite).
Il n'y a pas de prise, et les sauts ne sont pas autorisés ; on ne peut pas non plus se placer sur une case déjà occupée.

CHIFFRES (N ↑, E →, S ↓)

LETTRES (N ↑, O ←, E →)

LE PROGRAMME :
Les positions des chiffres et des lettres sont enregistrées dans une table N (5, 5) ; les lignes 30 à 70 définissent les valeurs de départ. Les indices 0 et 5 servent à délimiter le terrain de jeu.
Les lignes 100 à 140 dessinent le terrain de jeu et la ligne 170 affiche le joueur qui a le trait.
La ligne 190 contrôle le coup joué, de même que les boucles I et J (200 à 240), et le changement de joueur est assuré par la ligne 330.

POUR JOUER :
Le terrain de jeu sera affiché après la frappe de RUN et l'ordinateur attend le coup du joueur désigné.
On entrera d'abord le chiffre ou la lettre représentant le pion à déplacer, suivi d'une indication de direction sous la forme N (pour Nord), S (pour Sud), E (pour Est) ou O (pour Ouest).
On pourra ainsi faire 2E suivi de ENTER.
Si le coup est invalide, il ne sera pas exécuté, et le code du joueur restera à l'affichage ; s'il est autorisé, le déplacement sera effectué, et l'on obtient l'affichage correspondant au deuxième joueur.
Le premier qui aura réussi à faire disparaître ses 3 pièces aura gagné.

EXTENSIONS POSSIBLES :
On peut envisager d'étendre le jeu avec davantage de chiffres ou de lettres et d'améliorer les contrôles. On peut aussi définir une stratégie pour AMSTRAD.

```
10 REM A TRAVERS
20 DIM N(5,5):K=1
30 FOR I=0 TO 5
40 N(I,1)=ASC("0")+I:N(4,I)=ASC("?")+I
50 N(I,0)=1:N(0,I)=1:N(5,I)=1:N(I,5)=1
60 NEXT I
70 N(4,1)=0
80 MODE 0
90 FOR U=1 TO 4
100 FOR V=1 TO 4
110 LOCATE 4*U,4*V-2:IF N(V,U)=0 THEN PE
N 6:PRINT "." ELSE PEN(2-(N(V,U)<52)):PR
INT CHR$(N(V,U)):PEN 1
130 NEXT V
140 NEXT U
150 LOCATE 3,18:PEN 2+K
160 PRINT "JOUEUR AYANT LES":LOCATE 6,19
:IF K=0 THEN PRINT "LETTRES" ELSE  PRINT
 "CHIFFRES"
170 LOCATE 1,22:PRINT CHR$(20);"VOTRE CO
UP (";STRING$(-(K=0),"O");"NE";STRING$(-
(K=1),"S");")  ";
180 INPUT "",R$:R$=LEFT$(UPPER$(R$)+"   "
,2):L$=LEFT$(R$,1):R$=RIGHT$(R$,1)
190 IF (K=0 AND R$="S") OR (K=1 AND R$="
O") THEN PRINT CHR$(7);:GOTO 170
200 FOR I=1 TO 4
210 FOR J=1 TO 4
220 IF N(I,J)=ASC(L$) THEN 260
230 NEXT J
240 NEXT I
250 PRINT CHR$(7);:GOTO 150
260 A=I:B=J
270 I=I-(R$="S")+(R$="N")
280 J=J-(R$="E")+(R$="O")
290 IF (I=0 AND K=0) OR (J>4 AND K=1) TH
EN 320
300 IF N(I,J)<>0 THEN PRINT CHR$(7);:GOT
O 150
310 N(I,J)=ASC(L$)
320 N(A,B)=0
330 K=ABS(K-1)
340 GOTO 80
```

Calculatrice

REFERENCES :
Difficulté de programmation : très difficile.
Catégorie de jeu : simulation d'une calculatrice.
Difficulté du jeu : moyenne.

LE JEU :
Il s'agit de la simulation, en affichage grand format, du fonctionnement d'une petite calculatrice de poche, de type 4 opérations. Les applications pédagogiques de ce programme sont évidentes et son fonctionnement très simple à maîtriser.

LE PROGRAMME :
En ligne 15, la fonction VA (I$) est définie pour remédier (en partie) aux singularités de la fonction VAL du Basic AMSTRAD.
Les variables B$ et C$, des lignes 2Ø et 3Ø, contiennent les informations codées qui permettront le dessin des chiffres en grand format.
Les lignes 6Ø à 15Ø s'occupent de la gestion des entrées.
La boucle I (16Ø-19Ø) réalise la représentation, en grand format, des chiffres sur l'écran, au fur et à mesure des entrées.
Le point décimal et le signe − sont gérés par les lignes 2ØØ et 21Ø.
Le nombre total des symboles entrés est contrôlé par la ligne 22Ø.
Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 27Ø à 39Ø (l'automatisme est géré par la ligne 23Ø, dans le programme principal).
Les sous-programmes placés aux lignes 4ØØ-41Ø, 42Ø-43Ø, 44Ø-45Ø et 46Ø-47Ø effectuent l'opération indiquée, selon la valeur de Z (ligne 31Ø).
Chaque résultat est mémorisé en A (F-1) (ligne 33Ø).

POUR JOUER :
On devra lancer le programme par RUN.
Puis, on tape les chiffres du premier nombre à entrer qui vont s'afficher un par un, au fur et à mesure (on ne pourra entrer plus de 8 symboles, y compris le signe − et le point décimal). La frappe d'un signe opératoire laisse l'affichage intact, mais l'introduction du prochain chiffre ou symbole provoque l'effacement immédiat et l'affichage du nouveau symbole.
Il est à remarquer que les chiffres s'introduisent de gauche à droite, sens de l'écriture usuelle.
L'appareil accepte, comme signes opératoires les symboles +, −, *, / ; le signe = provoquera l'affichage du résultat. Les opérations sont enchaînées automatiquement, le résultat précédent étant pris en compte comme opérateur du calcul suivant.
Pour faire une autre opération, il faudra taper ESC deux fois, puis RUN.
Taper ESC deux fois pour sortir du programme.

EXTENSIONS POSSIBLES :
Protéger la saisie contre les entrées erronées (division par Ø ou plusieurs points décimaux à la suite).
On devra, tout d'abord, améliorer l'affichage du signe – (représenté, ici, par un trait), puis on pourra améliorer le dessin des chiffres (en utilisant, par exemple, le générateur de l'AMSTRAD), on peut aussi envisager une extension à plus de 8 chiffres, autoriser des opérations plus complexes (COS, SIN ...), ajouter des mémoires, etc.

```
10 REM CALCULATRICE
15 DEF FN VA(I$)=VAL(I$+STRING$(-(I$="-"
 OR I$="+"),"0"))
20 C$=STRING$(4,233)+STRING$(4,32)+STRIN
G$(2,233)+" "+CHR$(233)
30 B$="1444133333131211313144133121311214
411333314141141 31"
40 F=1:L=0:CLS
50 N=0:T=0:K=0:R$=""
60 A$=INKEY$:IF A$="" THEN 60
70 IF K=1 THEN N=0:R$="":K=0
80 IF A$="=" THEN 260
90 X=(1 AND A$="+")+(2 AND A$="-")+(3 AN
D A$="*")+(4 AND A$="/")
100 R$=R$+A$
110 IF X<>0 THEN 260
120 IF R$="-" THEN A$=R$:GOTO 200
130 IF A$<"." OR A$>"9" OR A$="/" THEN P
RINT CHR$(7);:GOTO 60
140 IF N=0 THEN CLS
150 IF A$="." OR A$="-" THEN 200
160 FOR I=1 TO 5
170 S=3*VAL(MID$(B$,I+5*FN VA(A$),1))-2
180 LOCATE N+1,I+4:PRINT MID$(C$,S,3)
190 NEXT I
200 IF A$="-" THEN CLS:LOCATE 2,7:PRINT
STRING$(2,233)
210 IF A$="." THEN LOCATE N+2,9:PRINT CH
R$(233)
220 N=N-4*(N<28)
230 IF T<>0 THEN T=T+1:IF T<=LEN(R$) THE
N 380
240 IF K=1 AND L<>1 THEN 50
250 T=0:GOTO 60
```

```
260 IF R$="-" AND A(F)=0 THEN 120
270 N=0:K=1
280 IF (A$="*" OR A$="/") AND FN VA(R$)=
0 THEN A(F)=1:GOTO 300
290 A(F)=FN VA(R$)
300 IF X<>0 THEN Z=X
310 ON Z GOSUB 400,420,440,460
320 IF A$<>"=" THEN F=F+1 ELSE F=2
330 A(F-1)=FN VA(R$)
340 IF FN VA(R$)=0 THEN A(F-1)=A(F)
350 R$=STR$(FN VA(R$))
360 IF LEFT$(R$,1)=" " THEN R$=RIGHT$(R$
,LEN(R$)-1)
370 T=1
380 A$=MID$(R$,T,1)
390 GOTO 140
400 R$=STR$(A(F-1)+A(F)):A(F)=FN VA(R$):
L=1
410 RETURN
420 IF F<>1 THEN R$=STR$(A(F-1)-A(F)):A(
F)=FN VA(R$):L=1
430 RETURN
440 IF F<>1 THEN R$=STR$(A(F-1)*A(F)):A(
F)=FN VA(R$):L=1
450 RETURN
460 IF F<>1 THEN R$=STR$(A(F-1)/A(F)):A(
F)=FN VA(R$):L=1
470 RETURN
```

Chasse au sous-marin

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes à bord d'un bâtiment de surface, chargé de traquer un sous-marin caché quelque part dans votre secteur ; vous devez vous déplacer jusqu'à venir au-dessus du sous-marin et tirer à une certaine profondeur ; mais, attention : le sous-marin, lui aussi, est mobile ; il ne coupera ses moteurs que lorsque vous serez exactement à sa verticale, pour ne pas être repéré.

LE PROGRAMME :
La boucle WHILE/WEND (3Ø-5Ø) saisit la dimension de la grille.
La boucle I (8Ø-14Ø) affiche le terrain.
La ligne 16Ø définit la position initiale du sous-marin ; la ligne 17Ø permet son déplacement, contrôlé par la ligne 18Ø (on reste dans les limites du terrain).
La ligne 2ØØ affiche votre position.
Les lignes 24Ø à 26Ø contrôlent votre déplacement.
La distance est calculée en ligne 27Ø, affichée en ligne 28Ø (dans une fenêtre # 1) et testée en lignes 29Ø et 3ØØ.
Le test de fin de partie est effectué à la ligne 33Ø.

POUR JOUER :
On lance le programme par RUN et on entre la dimension ; la grille de jeu s'affiche. Vous êtes au départ en position (Ø, Ø) et vous vous déplacez avec les flèches.
Après chaque déplacement, la nouvelle distance du sous-marin s'affiche ; à chaque coup le sous-marin se déplace. Si vous réussissez à vous placer à sa verticale, la distance Ø est affichée et il ne bouge plus.
Vous tirez en donnant une profondeur entière (entre 2 et 4, bornes comprises). Si vous le ratez, il se replace n'importe où, et on recommence ; si vous le coulez, vous obtenez l'affichage BRAVO et la partie s'arrête.
On peut arrêter à tout moment le programme en utilisant les touches ESC deux fois.

EXTENSIONS POSSIBLES :
Il est possible de prévoir une amélioration du graphisme, d'améliorer les possibilités de déplacement du chasseur (un peu lent, dans cette version), de prévoir des parcours plus sophistiqués pour le sous-marin, de mettre un compteur de temps...

```
10 REM CHASSE AU SOUS-MARIN
20 RANDOMIZE(TIME)
30 WHILE H<1 OR H>10
40 PRINT:INPUT "DIMENSION DE LA GRILLE :
";H
50 WEND
60 CLS
70 WINDOW #1,4+2*(H+1),39,H+2,H+5:PEN #1
,0:PAPER #1,1
80 FOR I=0 TO H
90 PEN 1:LOCATE 2*I+4,1:PRINT I
100 LOCATE 1,2*I+3:PRINT I:PEN 2
110 FOR J=0 TO H
120 LOCATE 2*J+5,2*I+3:PRINT "."
130 NEXT J
140 NEXT I
150 I=1:X=0:Y=0
160 V=INT(RND*H):U=INT(RND*H)
170 U=U+I:V=V+I
180 IF (U<0 OR U>H OR V<0 OR V>H) THEN I
=-I:GOTO 170
190 LOCATE 2*Y+3,2*X+3:IF Y<>0 THEN PRIN
T "."
200 LOCATE 2*Y+5,2*X+3:PEN 3:PRINT CHR$(
255):PEN 2
210 LOCATE 2*Y+7,2*X+3:IF Y<>H THEN PRIN
T "."
220 LOCATE 2*Y+5,2*X+1:IF X<>0 THEN PRIN
T "."
230 LOCATE 2*Y+5,2*X+5:IF X<>H THEN PRIN
T "."
240 A$=INKEY$:IF A$="" THEN 240
250 Y=Y-(A$=CHR$(&F3) AND Y<H)+(A$=CHR$(
&F2) AND Y>0)
260 X=X-(A$=CHR$(&F1) AND X<H)+(A$=CHR$(
&F0) AND X>0)
270 D=ROUND(SQR(ABS(X-V)↑2+ABS(Y-U)↑2))
280 CLS #1:LOCATE #1,14-H,1:PRINT #1,"DI
STANCE":PRINT #1,USING SPACE$(16-H)+"##"
;D
290 IF D=0 THEN 320
300 IF D<3 THEN 170
310 GOTO 190
320 LOCATE #1,13-H,3:PRINT #1,"PROFONDEU
R":LOCATE #1,18-H,4:INPUT #1,"",D
330 IF D=INT(RND*3+2) THEN LOCATE 1,25:P
RINT "BRAVO!":SOUND 1,478:SOUND 1,379:SO
UND 1,319:END
340 TE=0:WHILE TE<20:SOUND 1,TE,1:TE=TE+
1:WEND
350 GOTO 160
```

Niveau 5

Nom du programme	*Thèmes et particularités*
Horloge 1	Précise, facile à régler, et très colorée.
Horloge 2	Ne jetez pas votre vieille télévision : transformez-la en horloge.
Combat aérien	Pas si facile de descendre un appareil qui bouge tout le temps.
Affichage publicitaire	Si vous ne savez pas quoi mettre dans votre vitrine.
Le jeu du missile	Une mission bien périlleuse (un seul joueur).
Sous-marin	Un classique des jeux vidéo (1 joueur).
Tic Tac Toe	Un classique, mais pas sans intérêt (deux joueurs).
Smokey	Fumez sans risque, avec ce programme original pour un joueur.
Course de voitures	Un jeu de réflexe pas trop difficile, pour un joueur.
Danger spatial	Serez-vous assez adroit pour éviter toutes les météorites.
Diago	Etes-vous vraiment sûr de vos réflexes ?
Oeufs	Ne laissez pas faire la pondeuse diabolique (un joueur).
Taquin de lettres	Un proverbe est caché dedans : à vous de le découvrir.
Cibles	Un peu d'adresse, beaucoup de chance : très visuel.
Gouttes	Un véritable gag visuel. A exploiter.
Obstacles	Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur.
Captures dans l'espace	Votre mission serait facile, sans ces damnées météorites.
Anomalie	Exercez votre perspicacité dans la reconnaissance des symboles.
Plusieurs en un	Un terrain de jeu et les pions, pour deux joueurs.
Parcours du cavalier	Très spectaculaire, et instructif à plus d'un titre.
Othello	Pour deux joueurs ; il retourne les pions à votre place.

Dames	Pour deux joueurs : classique
Quatre à quatre	Un jeu de réflexion un peu difficile, pour deux joueurs.
Echecs	Un programme assez simplifié, pour deux joueurs.
Mur de briques	Un grand classique dans une version très complète.
Le lézard d'Alice	Beaucoup de hasard, mais très spectaculaire.

Les techniques particulières à l'AMSTRAD

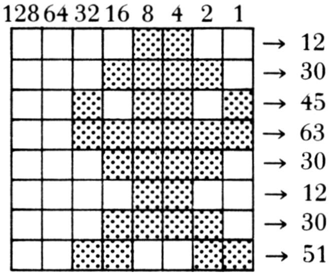
UNE INSTRUCTION DE CRÉATION DE FORMES : L'INSTRUCTION SYMBOL

Le Basic AMSTRAD permet de redéfinir les caractères affectés aux touches du clavier. L'ordre SYMBOL AFTER indique à partir de quel code ASCII on désire commencer.

Le caractère de code N est créé par l'ordre :

SYMBOL N, 12, 3Ø, 45, 63, 3Ø, 12, 3Ø, 51

Il a été défini ainsi :



L'ordre TAG, en indiquant que les impressions de texte doivent être effectuées à la position du curseur graphique, permet de mettre le caractère en n'importe quel point de l'écran par un simple "PRINT CHR$(N) ;" (ne pas oublier le point-virgule qui évite l'apparition de caractères parasites).

LES INTERRUPTIONS

A certains instants, l'ordinateur peut abandonner le programme qu'il exécute, pour accomplir une autre tâche, avant de reprendre son travail antérieur.
Contrôlés par l'horloge interne d'AMSTRAD, ces sous-programmes sont utiles lorsque le temps joue un rôle.
L'ordre EVERY appelle un sous-programme à intervalle régulier (par exemple pour simuler le battement d'une pendule).
Il est possible de déconnecter temporairement l'interruption par un ordre DI. EI la rétablira.

IDENTIFICATION D'UN OBJET SUR L'ÉCRAN : TEST

La fonction TEST permet de faire la lecture directe de la couleur d'un point de l'écran aux coordonnées X et Y.

Horloge 1

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : simulation.
Difficulté du jeu : aucune.

LE JEU :
Le programme simule le fonctionnement d'une horloge digitale utilisant les caractères agrandis d'AMSTRAD. Il présente la particularité de faire appel, pour une bonne précision sur l'heure, à l'horloge interne d'AMSTRAD.

LE PROGRAMME :
La ligne 60 permet d'entrer l'heure en tapant d'abord les heures, puis les minutes, puis les secondes. Le sous-programme 1000-1080 est utilisé pour définir les zones de saisie, contrôler les entrées et valider la saisie.
La boucle WHILE/WEND (110-200) s'occupe des heures et celle comprise entre les lignes 120-180, des minutes.
La boucle WHILE/WEND (130-160) affiche les secondes. Le sous-programme (2000-2010) incrémente les secondes (la ligne 100 autorise le comptage des secondes dans le sous-programme (2000-2010) appelé grâce à un ordre EVERY en ligne 20).
Le sous-programme (3000-3010) assure l'impression des heures, minutes et secondes sur deux chiffres.

POUR JOUER :
Faire RUN : la question « Entrez l'heure sous la forme HH : MM : SS » est affichée, et on tape l'heure, puis les minutes, et les secondes.
Le comptage du temps démarre dès la frappe de ENTER.
On arrête en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
On peut envisager d'afficher les secondes, ajouter une possibilité de réveil, prévoir un agenda automatique, faire apparaître un coucou chantant au moment du passage de l'heure, etc.

```
10 REM HORLOGE 1
20 EVERY 50 GOSUB 2000:DI
30 CLS
40 PRINT:PRINT "ENTREZ L'HEURE SOUS LA F
ORME HH:MM:SS"
50 LOCATE 12,4:PRINT "--:--:--":PEN 3
60 FOR N=0 TO 2:LOCATE 12+3*N,4:GOSUB 10
00:T$=T$+R$:NEXT N
70 FOR I=1 TO 3:X(I)=VAL(MID$(T$,2*I-1,2
)):GOSUB 3000:NEXT I
80 MODE 0
90 MOVE 11*16,6*32+8:DRAWR 0,32,2:DRAWR
9*32,0,2:DRAWR 0,-32,2:DRAWR -9*32,0,2
100 EI
110 WHILE X(1)<24
120 WHILE X(2)<60
130 WHILE X(3)<60
140 I=3:GOSUB 3000
150 LOCATE 7,12:PRINT X$(1);" ";X$(2);"
";X$(3)
160 WEND
170 X(3)=0:X(2)=X(2)+1:PEN 1+X(2) MOD 12
:I=2:GOSUB 3000
180 WEND
190 X(2)=0:I=2:GOSUB 3000:X(1)=X(1)+1:I=
1:GOSUB 3000
200 WEND
210 FOR I=1  TO 3:X(I)=0:GOSUB 3000:NEXT
 I
220 GOTO 110
1000 R$="":S=0
1010 A$=INKEY$:IF A$="" THEN 1010
1020 IF (N=0 AND S=0 AND A$<>"0" AND A$<
>"1" AND A$<>"2") THEN 1010
1030 IF ((N=1 OR N=2) AND S=0 AND A$<>"0
" AND VAL(A$)>=6) THEN 1010
1040 IF (S=1 AND A$<>"0" AND VAL(A$)=0)
THEN S=S-1:GOTO 1010
1050 R$=R$+A$:PRINT A$;:S=S+1
1060 IF S<2 THEN 1010
1070 IF N=0 AND VAL(R$)>24 THEN RUN
1080 RETURN
2000 X(3)=X(3)+1:PRINT CHR$(7);
2010 RETURN
3000 X$(I)="0"+RIGHT$(STR$(X(I)),LEN(STR
$(X(I)))-1):X$(I)=RIGHT$(X$(I),2)
3010 RETURN
```

Horloge 2

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
Transformer votre écran de télévision en horloge numérique à affichage grand format, grâce à ce programme qui vous permettra d'animer une vitrine, de faire une horloge fantaisie, et qui pourra se prêter à bien d'autres usages.

LE PROGRAMME :
La ligne 3Ø vous permettra d'entrer l'heure sous forme d'un entier de 4 chiffres (par exemple : 1Ø24 signifera 1Ø heures 24 minutes).
Le sous-programme 1ØØØ-1Ø3Ø est exécuté toutes les secondes, tandis que le sous-programme 2ØØØ-221Ø l'est toutes les minutes.
Les lignes 2Ø7Ø-219Ø réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 218Ø).
Les lignes 2ØØØ à 2Ø4Ø assurent la progression de l'heure.
La ligne 1ØØØ effectue l'affichage de deux points clignotants pour les secondes.

POUR JOUER :
Dès le lancement par RUN, le programme se place en attente de l'heure ; on tape et on attend le top pour appuyer sur ENTER.
L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur la touche ESC deux fois de suite.
L'heure sera entrée sous forme d'un entier ayant impérativement 4 chiffres, pour 5 heures 6 minutes, on veillera à taper Ø5Ø6 suivi de ENTER.

EXTENSIONS POSSIBLES :
Il sera intéressant de modifier le programme pour en faire un réveil, avec alarme ; on peut aussi améliorer le graphisme, introduire la couleur.

```
10 REM HORLOGE 2
20 CLS
30 INPUT "HEURE ( FORME XXXX )";A$
40 GOSUB 2050
50 EVERY 50,1 GOSUB 1000
60 EVERY 50*60,2 GOSUB 2000
70 GOTO 70
1000 LOCATE 20,4:PAPER H:PRINT " ":LOCAT
E 20,6:PRINT " ":PAPER 2
1010 SOUND 1,473,1
1020 H=ABS(H-3)
1030 RETURN
2000 A=1+VAL(A$)
2010 IF A=2360 THEN A=0
2020 A$="000"+RIGHT$(STR$(A),LEN(STR$(A)
)-1)
2030 A$=RIGHT$(A$,4)
2040 IF RIGHT$(A$,2)="60" THEN A$=RIGHT$
(STR$(VAL(LEFT$(A$,2))+1),2)+"00"
2050 CLG 0:PAPER 2
2060 K=1
2070 WHILE K<5
2080 A=VAL(MID$(A$,K,1))
2090 IF A<>1 AND A<>4 THEN LOCATE 6*K+4,
1:PRINT "    "
2100 FOR J=1 TO 5
2110 IF A=0 OR (A>3 AND A<>7) THEN LOCAT
E 6*K+4,J:PRINT " "
2120 IF A<>5 AND A<>6 THEN LOCATE 6*K+6,
J:PRINT " "
2130 IF A MOD 2=0 AND A<>4 THEN LOCATE 6
*K+4,J+4:PRINT " "
2140 IF A<>2 THEN LOCATE 6*K+6,J+4:PRINT
 " "
2150 NEXT J
2160 IF A>1 THEN LOCATE 6*K+5,5:PRINT "
 "
2170 IF A<>1 AND A<>4 AND A<>7 THEN LOCA
TE 6*K+4,9:PRINT "    "
2180 K=K+1
2190 WEND
2200 SOUND 1,478 :SOUND 1,253:SOUND 1,35
8
2210 RETURN
```

Combat aérien

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexe (1 joueur).
Difficulté du jeu : difficile.

LE JEU :
Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.
Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.
Vous tirez sur lui en frappant COPY, et vous voyez vos projectiles qui convergent sur lui. Mais il continue à se déplacer, même pendant votre tir, et il n'est pas du tout évident que vous réussirez à l'avoir.
La partie continuera à se dérouler jusqu'à ce que vous l'ayez touché (affichage de GAGNE).

LE PROGRAMME :
La ligne 7Ø dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 9Ø et 1ØØ.
Les lignes 12Ø à 13Ø vous permettent le pilotage de votre appareil.
La ligne 17Ø dessine le viseur de votre appareil.
Le sous-programme, gérant le tir, va de la ligne 2ØØ à la ligne 25Ø et le test de fin de partie est effectué en ligne 24Ø.
La ligne 18Ø regarde si vous avez tiré ou non.
Le bruit de la mitrailleuse est provoqué par l'appel du sous-programme 26Ø, 27Ø pendant le temps du tir, grâce à l'ordre EVERY.

POUR JOUER :
Dès que le programme a été lancé par l'ordre RUN, vous devez manœuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.
Pour cela, vous disposez des flèches de déplacement (touche ↑, ↓, →, ←) et vous pouvez tirer à tout moment en pressant la touche COPY.
L'appareil ennemi peut continuer à se déplacer de manière aléatoire, même lorsque vous avez tiré.
La partie se terminera avec l'affichage de GAGNE, si vous touchez l'avion ennemi.
On peut aussi arrêter à tout moment en frappant ESC deux fois.

EXTENSIONS POSSIBLES :
Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait envisager de stabiliser la position de l'avion ennemi pendant le tir.
On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.

```
10 REM COMBAT AERIEN
20 RANDOMIZE(TIME)
30 EVERY 4,1 GOSUB 260:DI
40 CLS
50 A=5:B=15:AV$=CHR$(242)+CHR$(198)+CHR$
(243)
60 N=21
70 LOCATE B,A:CALL &BD19:PRINT AV$
80 U=A:V=B
90 A=A+2*INT(RND*2)-1
100 B=B+2*INT(RND*2)-1
110 A$=INKEY$
120 B=B-2*(A$=CHR$(&F3))+2*(A$=CHR$(&F2)
)
130 A=A-2*(A$=CHR$(&F1))+2*(A$=CHR$(&F0)
)
140 IF B<1 OR B>38 THEN B=ABS(B-37)
150 IF A<1 OR A>24 THEN A=ABS(A-23)
160 LOCATE V,U:CALL &BD19:PRINT "   "
170 LOCATE 18,9:PEN 2:PRINT CHR$(150);CH
R$(158);CHR$(156):LOCATE 18,10:PRINT CHR
$(151);CHR$(144);CHR$(157):LOCATE 18,11:
PRINT CHR$(147);CHR$(155);CHR$(153):PEN
1
180 IF A$=CHR$(&E0) OR N<21 THEN GOSUB 2
00
190 GOTO 70
200 EI:PEN 3:LOCATE 29-N,N:PRINT CHR$(20
4):LOCATE N+9,N:PRINT CHR$(205):PEN 1
210 LOCATE 29-N,N:PRINT " ":LOCATE N+9,N
:PRINT " "
220 N=N-1
230 IF N>9 THEN RETURN
240 IF ((ABS(10-A))<=1) AND ((20-B)<=4 A
ND B<=20) THEN LOCATE B,A:PEN 3:PRINT AV
$:PAPER 2:PEN 3:LOCATE 17,23:PRINT "GAGN
E!":PAPER 0:END
250 N=21:DI:RETURN
260 SOUND 1,271,1,5
270 RETURN
```

Affichage publicitaire

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : affichage publicitaire.
Difficulté du jeu : aucune.

LE JEU :
Un texte quelconque devra être introduit en début de programme. Il pourra être relativement long (mais inférieur à 254 caractères) ; on choisira ensuite le sens et la vitesse de défilement, à la suite de quoi le texte va apparaître (une seule ligne à la fois) et commencera à défiler, à la vitesse choisie, et dans le sens choisi.

LE PROGRAMME :
Le texte sera introduit lettre par lettre aux lignes 7Ø à 1ØØ.
L'affichage sera en grand format, grâce à la ligne 16Ø.
Le défilement vers la gauche est préparé à la ligne 19Ø, et le défilement vers la droite à la ligne 2ØØ.
Le texte sera affiché en ligne 18Ø, et la durée de l'affichage (vitesse de défilement) est choisie par la boucle 21Ø-24Ø.
La ligne 23Ø permettra de changer, si besoin est, les conditions du défilement.
Un ordre EVERY, suspendu pendant la saisie des données, permet de changer régulièrement la couleur des impressions (sous-programme 26Ø-27Ø).

POUR JOUER :
On lancera le programme par RUN, et à l'affichage de la question ENTREZ VOTRE TEXTE, on introduira le texte à faire défiler, suivi de ENTER.
On obtient ensuite l'affichage DEFILEMENT DROITE (D) OU GAUCHE (G) auquel on répondra en tapant, soit D, soit G.
Puis à la question VITESSE, on répondra en introduisant un grand nombre pour une vitesse rapide, et un petit pour une vitesse lente (par exemple, de Ø à 4Ø ou plus).
Le texte commencera à défiler dès que l'on aura frappé ENTER.
Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur la touche Ø pour retrouver les questions SENS et VITESSE. On tapera ESC deux fois pour interrompre le programme.

EXTENSIONS POSSIBLES :
On pourra introduire de la couleur, réaliser un affichage ondulant, faire remonter les lignes sur l'écran, etc.

```
10 REM AFFICHAGE PUBLICITAIRE
20 EVERY 250 GOSUB 260:DI
30 CLS
40 B$=""
50 PRINT "ENTREZ VOTRE TEXTE :"
60 PRINT
70 A$=INKEY$:IF A$="" THEN 70
80 IF A$=CHR$(13) THEN B$=B$+" ":GOTO 11
0
90 B$=B$+A$
100 PEN 3:PRINT A$;:GOTO 70
110 PRINT:PRINT:PEN 1:PRINT "DEFILEMENT
DROITE (D) OU A GAUCHE (G) ";:PEN 2:INPU
T"",S$:PEN 1
120 PRINT
130 PRINT "VITESSE (DE 0 A 50) :";:PEN 2
:INPUT "",V:PEN 1
140 V=51-V
150 EI
160 MODE 0
170 T=LEN(B$)
180 LOCATE 1,12:PRINT " "+MID$(B$,1,18)+
" "
190 IF LEFT$(S$,1)="G" THEN B$=RIGHT$(B$
,T-1)+LEFT$(B$,1)
200 IF LEFT$(S$,1)="D" THEN B$=RIGHT$(B$
,1)+LEFT$(B$,T-1)
210 FOR J=0 TO 10*V
220 C$=INKEY$
230 IF C$="O" THEN MODE 1:DI:GOTO 110
240 NEXT J
250 GOTO 180
260 C=(C+1) MOD 25:INK 1,1+C
270 RETURN
```

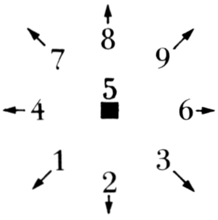
Le jeu du missile

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : contre AMSTRAD.
Difficulté du jeu : moyenne.

LE JEU :
AMSTRAD est en alerte : vous êtes aux commandes d'un avion chargé d'une mission d'espionnage sur son territoire ; votre but est de survoler une base ennemie (représentée par un caractère agrandi), et de revenir sain et sauf sur votre base ; mais, dès qu'AMSTRAD vous aura repéré, il va envoyer à votre poursuite, un missile très perfectionné qui n'aura de cesse de vous détruire.
Votre avion peut se déplacer dans toutes les directions, grâce aux touches de 1 à 9 ; réussirez-vous à remplir votre mission ?

LE PROGRAMME :
La ligne 5Ø dessine la base ennemie à survoler (sous forme d'un H agrandi).
La ligne 6Ø dessine votre base de départ.
Les lignes 8Ø et 9Ø choisissent l'instant de départ du missile.
La ligne 1ØØ dessine votre avion.
Les lignes 12Ø et 13Ø définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 22Ø, 23Ø, 25Ø et 26Ø.
La ligne 14Ø assure le départ du missile qui est dessiné par la ligne 15Ø et dont le déplacement est géré par les lignes 16Ø et 17Ø.
Les lignes 18Ø et 2ØØ contrôlent la fin de partie.

POUR JOUER :
Lancer le programme par RUN.
Le terrain de jeu s'affiche et vous pouvez déplacer votre appareil en utilisant les touches de 1 à 9, avec la convention suivante :



Vous devez lui faire contourner le territoire ennemi (si le missile touche ce territoire, la partie s'arrête).
Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête.
Taper ESC deux fois pour arrêter la partie.

EXTENSIONS POSSIBLES :
Il sera sans doute nécessaire d'améliorer les divers contrôles du programme (l'avion a-t-il survolé la base ennemie, est-il revenu à sa propre base...). On peut aussi prévoir divers niveaux de jeu en faisant, par exemple, varier la vitesse du missile, prévoir l'utilisation de leurres par l'avion, l'explosion du missile au bout d'un certain temps, etc.

```
10 REM LE JEU DU MISSILE
20 RANDOMIZE(TIME)
30 EVERY 10 GOSUB 280:DI
40 CLS:M=0
50 LOCATE 18,6:PEN 3:PRINT CHR$(141);CHR
$(142);CHR$(10);STRING$(2,8);CHR$(135);C
HR$(139)
60 PEN 2 :LOCATE 1,21:PRINT STRING$(4,12
7):LOCATE 1,22:PRINT STRING$(4,127):PEN
1
70 U=34:V=16:X=2:Y=20
80 S=INT (RND*V+3)
90 S=S-1
100 LOCATE X,Y:PRINT CHR$(239);
110 A$=INKEY$:IF A$="" THEN 110
120 A=VAL(A$):F=SGN(6.5-A)+2*(A=4)
130 ON (A AND A<5)+((10-A) AND A>5) GOSU
B 220,230,250,260
140 IF S<>0 THEN 90 ELSE EI
150 LOCATE U,V:PEN 3:PRINT CHR$(252);:PE
N 1
160 U=U-SGN(U-X)
170 V=V-SGN(V-Y)
180 IF (U=X AND U=Y) THEN LOCATE 1,23:PR
INT "TOUCHE...":INK 0,3,0:SOUND 1,319:SO
UND 1,379:SOUND 1,478:FOR I=1 TO 500:NEX
T I: INK 0,1:END
190 IF X>=20 AND Y<=5 THEN M=1
200 IF M=1 AND X<=4 AND Y=20 THEN LOCATE
 1,23:PRINT "MISSION ACCOMPLIE":SOUND 1,
478:SOUND 1,379:SOUND 1,319:END
210 GOTO 100
220 X=X-F
230 Y=Y+F
240 RETURN
250 GOSUB 230
260 X=X+F
270 RETURN
280 SOUND 1,1000,2,2,0,0,1
290 RETURN
```

Sous-marin

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : facile.

LE JEU :
Vous manœuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.
Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.
Votre score sera affiché en permanence, sur le centre de l'écran (rapport nombre de coups au but sur nombre de sous-marins apparus).

LE PROGRAMME :
La ligne 7Ø permet à votre AMSTRAD de choisir la profondeur du sous-marin.
La ligne 9Ø dessine votre bâtiment.
La ligne 1ØØ dessine la surface de l'eau.
La ligne 12Ø dessine le sous-marin.
Le tir est effectué par un programme spécifique placé aux lignes 2ØØ à 24Ø.
Le déplacement (très rapide) de votre navire est réalisé par les lignes 17Ø et 18Ø.
Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 14Ø.

POUR JOUER :
Taper RUN pour commencer la partie.
Le bâtiment sera déplacé grâce aux flèches droite et gauche.
La touche Ø autorise le tir.
Le navire, se déplaçant trois fois plus vite que les sous-marins, il sera possible de toucher plusieurs fois le même bâtiment.
Pour terminer la partie, il conviendra de faire ESC deux fois ou d'attendre 99 sous-marins.

EXTENSIONS POSSIBLES :
Il sera assez élémentaire d'améliorer la qualité graphique du jeu, d'utiliser la couleur et les sons, de faire apparaître plusieurs sous-marins se déplaçant à des vitesses différentes, éventuellement même, de permettre aux sous-marins de tirer...

```
10 REM SOUS-MARIN
20 RANDOMIZE(TIME):EVERY 100 GOSUB 250
30 INK 0,0:INK 1,6:INK 2,18:INK 3,2
40 P=0:C=16:S=5
50 CLS
60 LOCATE 1,S:PRINT STRING$(5," ")
70 T=37:S=INT(RND*8+5):P=P+0.01
80 X=2
90 PEN 2:LOCATE C,3:PRINT "    ";CHR$(213
);CHR$(142);CHR$(141);CHR$(212);"    "
100 LOCATE 1,4:PEN 3:PRINT STRING$(40,CH
R$(196))
110 PAPER 1:PEN 2:LOCATE 15,19:PRINT USI
NG "SCORE ###.##";P:PAPER 0
120 PEN 2:LOCATE T,S:PRINT CHR$(154);CHR
$(231);CHR$(154);" "
130 IF X=2 THEN Y=C+2
140 IF S=X AND (T+1=Y OR T+2=Y) THEN P=P
+1:LOCATE T,S:PRINT "***":SOUND 1,456,20
,3,0,0,1
150 T=T-1:IF 100*(P-INT(P))>=99 THEN LOC
ATE 1,21:PRINT "C'EST FINI .............
>...............":SOUND 1,478:SOUND 1,42
6:SOUND 1,379:END
160 IF T<1 THEN LOCATE Y,X:PRINT " ":GOT
O 60
170 A$=INKEY$
180 C=C+3*((A$=CHR$(&F2) AND C>1)-(A$=CH
R$(&F3) AND C<33))
190 IF A$<>"O" AND X=2 THEN 90
200 LOCATE Y,X:IF X=2 THEN PEN 2 ELSE PE
N 1
210 PRINT " ":X=X+1
220 IF X>S THEN 80
230 LOCATE Y,X:PRINT CHR$(252)
240 GOTO 90
250 SOUND 1,200,8:RETURN
```

Tic tac toe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : facile.

LE JEU :
Un grand classique des jeux d'écoliers, proposé ici dans une version ne comportant pas le jeu machine ; chaque joueur devra placer à son tour un O ou un X (ici un cœur ou un carreau) dans une case libre de la grille de jeu, et le premier qui réussit un alignement de 3 symboles identiques a gagné (l'alignement peut être horizontal, vertical, ou suivant l'une des deux diagonales).

LE PROGRAMME :
Le terrain de jeu est dessiné à l'aide de la boucle I (5Ø-7Ø).
La ligne 9Ø s'occupe d'afficher le symbole cœur ou carreau correspondant au joueur qui a le trait.
Un symbole clignotant (ligne 12Ø) est affiché en haut et à gauche du terrain, même si la case est déjà occupée ; les lignes 15Ø et 16Ø permettent de déplacer le symbole sur le terrain de jeu.
La ligne 17Ø vérifie si la case où l'on veut jouer est libre, et si l'on désire jouer.
La ligne 18Ø enregistre le coup, et la ligne 19Ø réalise le changement de joueur.

POUR JOUER :
Le terrain de jeu sera affiché dès la frappe de RUN.
Un symbole clignotant apparaît alors en haut et à gauche de l'écran ; le joueur peut le déplacer sur le terrain de jeu à l'aide des 4 flèches (touches ←, → et ↑, ↓).
Lorsqu'il arrive sur la case où il désire jouer, il suffit qu'il appuie sur la touche COPY, le coup sera refusé si la case n'est pas libre.
Si la case est libre, le symbole qui correspond à ce joueur est inscrit, et on change de joueur.
Lorsque la partie est finie, il convient de taper ESC deux fois pour sortir du programme. On pourra ensuite faire une autre partie avec RUN.

EXTENSIONS POSSIBLES :
On pourrait réaliser après chaque coup un contrôle pour vérifier si l'un des joueurs a gagné. On peut aussi réaliser un jeu machine mais il faudrait éviter qu'il soit par trop systématique (par exemple, la machine jouant toujours son premier coup sur la case centrale).

```
10 REM TIC TAC TOE
20 MODE 0
30 INK 0,18:INK 1,16:INK 2,26:INK 3,6:BO
RDER 16
40 CLG 1
50 FOR I=1 TO 9
60 PEN 0:LOCATE 6,I+1:PRINT STRING$(9,14
3)
70 NEXT I
80 K=0
90 LOCATE 6,18:PAPER 6:PEN 2+K:PRINT "JO
UEUR  "+CHR$(228-K):PAPER 0
100 I=1:J=1
110 C=TEST ((3*(I+1))*32+16,(25-(3*J))*1
6+8)
120 PEN 2+K:LOCATE 1+3*(I+1),3*J:PRINT C
HR$(143)
130 LOCATE 1+3*(I+1),3*J:IF C<>0 THEN PE
N C:PRINT CHR$(230-C) ELSE PRINT " "
140 A$="":A$=INKEY$
150 I=I-(A$=CHR$(&F3) AND I<3)+(A$=CHR$(
&F2) AND I>1)
160 J=J-(A$=CHR$(&F1) AND J<3)+(A$=CHR$(
&F0) AND J>1)
170 IF C<>0 OR A$<>CHR$(&E0) THEN 110
180 LOCATE 1+3*(I+1),3*J:PEN 2+K:PRINT C
HR$(228-K)
190 K=ABS(K-1)
200 GOTO 90
```

Smokey

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Une colonne de fumée jaune apparaît sur l'écran. Le joueur, représenté par une tête, devient visible sur la droite. Il doit dans un premier temps se positionner sur la colonne de fumée, puis à partir de là, il va automatiquement avaler la fumée, à condition qu'il réussisse à se maintenir sur sa trace.

LE PROGRAMME :
La ligne 6Ø fait apparaître la fumée au bas de l'écran ; la ligne 7Ø permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 9Ø fait monter la colonne.
Le joueur apparaît grâce à la ligne 11Ø, et son déplacement est effectué par les lignes 16Ø à 18Ø.
Les lignes 23Ø à 25Ø se chargent de l'affichage du résultat.

POUR JOUER :
On doit lancer l'exécution du programme par RUN.
La fumée commence à s'élever, tandis que le joueur, apparu sous forme d'une tête rouge dans un coin de l'écran, va se déplacer sur elle à l'aide des flèches. Dès qu'il arrive sur la fumée, il va automatiquement commencer à l'avaler, et devra essayer de suivre la colonne sans dévier (grâce aux flèches). Il gagne s'il arrive en bas de l'écran avant que la fumée n'arrive en haut.
L'affichage final indique alors BRAVO ou PERDU, ainsi que le score obtenu.

EXTENSIONS POSSIBLES :
On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.
On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.

```
10 REM SMOKEY
20 RANDOMIZE(TIME)
30 S=0:X=24:Y=1+INT(RND*40)
40 U=1:V=17
50 CLS
60 LOCATE Y,X:PRINT CHR$(228)
70 Y=Y-(RND<0.5 AND Y<24)+(RND<0.5 AND Y
>1)
80 T=1
90 X=X-T
100 IF X=0 THEN 230
110 PEN 3:LOCATE V,U:PRINT CHR$(224);:PE
N 1
120 IF U>=24 THEN 230
130 IF T<>0 THEN FOR I=0 TO 300:A$=INKEY
$:IF A$<>"" THEN 150
140 IF T<>0 THEN NEXT I
150 LOCATE V,U:PRINT " "
160 U=U-(A$=CHR$(&F1) AND T=1 AND X<23)
170 U=U-(T=0)
180 V=V-(A$=CHR$(&F3) AND V<40)+(A$=CHR$
(&F2) AND V>1)
190 C=TEST((V-1)*16+8,(25-U)*16+8)
200 IF C<>1 THEN 60
210 T=0:S=S+1
220 GOTO 110
230 LOCATE 12,12:IF U>=24 THEN PRINT "
BRAVO":SOUND 1,478:SOUND 1,379:SOUND 1,3
19:GOTO 250
240 FOR I=1 TO 6:SOUND 1,278,10,6*(I MOD
 2):NEXT I:PRINT "PERDU....."
250 LOCATE 12,14:PRINT USING "SCORE : ##
";S
```

Course de voitures

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (1 joueur).
Difficulté du jeu : moyenne.

LE JEU :
Vous circulez sur une route très sinueuse, ne comportant aucun obstacle, et vous devez essayer de la suivre le plus longtemps possible. La trace de votre parcours reste affichée en permanence, et votre score est affiché dès que vous avez un accident.

LE PROGRAMME :
L'impression de la route est assurée par la boucle I (7Ø-1ØØ).
La ligne 11Ø fait avancer la route, et la ligne 12Ø imprime la voiture.
La ligne 15Ø contrôle son mouvement, et la ligne 16Ø détermine si la route fait ou non des virages.
La ligne 18Ø regarde si l'on a touché le bord de la route.
La ligne 19Ø incrémente le score.
La ligne 23Ø affiche le temps avant l'accident.

POUR JOUER :
Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par une voiture, apparaît.
Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches → et ← (flèches de déplacement du curseur). Il vous faudra éviter le plus longtemps possible de toucher une bordure noire, ce qui provoquera l'arrêt du programme et l'affichage de la mention ACCIDENT, suivie de l'indication sur le score réalisé.
On pourra arrêter le déroulement de la partie à tout moment en pressant 2 fois sur la touche ESC.

EXTENSIONS POSSIBLES :
On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.
On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.
On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.

```
10 REM COURSE DE VOITURES
20 RANDOMIZE(TIME)
30 SYMBOL AFTER 240
40 SYMBOL 240,93,127,85,20,93,127,93,8
50 T=TIME:CLS:PEN 1
60 X=24:Y=11:A=5:B=10:S=0
70 FOR I=0 TO 6
80 LOCATE Y+I+1,X+1:PRINT CHR$(143);
90 IF I<2 OR I>4 THEN PEN 2:LOCATE Y+I+1
,X+1:PRINT CHR$(207);:PEN 1
100 NEXT I
110 PRINT:PRINT
120 LOCATE B,A:PEN 3:PAPER 0-(N>18):PRIN
T CHR$(240):PAPER 0:PEN 2
130 FOR I=1 TO 10:A$=INKEY$:IF A$="" THE
N NEXT I
140 C=B
150 B=B-(A$=CHR$(&F3) AND B<40)+(A$=CHR$
(&F2) AND B>1)
160 Y=Y-(-2*RND AND Y<25)-(2*RND AND Y>3
)
170 IF N<19 THEN N=N+1:GOTO 200
180 IF TEST((B-1)*16+8,(25-A-1)*16+8)<>1
 THEN 220
190 S=S+1:PAPER 1
200 LOCATE C,A:PRINT " "
210 PAPER 0:GOTO 70
220 LOCATE 20,12:PAPER 2:PEN 3:PRINT USI
NG "ACCIDENT. SCORE ###";S
230 LOCATE 20,14:PRINT USING "TEMPS :###
";ROUND ((TIME-T)/300)
240 SOUND 1,478:SOUND 1,426:SOUND 1,379
250 PAPER 0:PEN 1
```

Danger spatial

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe, pour un joueur.
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes aux commandes d'un engin spatial qui vient de pénétrer dans un nuage de météorites. Vous devez essayer de franchir le nuage sans heurter une seule météorite. Votre score sera affiché en fin de partie, lorsque vous entrez en collision avec une météorite.

LE PROGRAMME :
La position de chaque météorite est déterminée par les lignes 6Ø et 8Ø ; elle est imprimée par les lignes 7Ø et 9Ø.
L'avance des météorites est réalisée par la ligne 1ØØ.
Les lignes 1ØØ et 11Ø vérifient si vous avez ou non heurté une météorite.
La ligne 12Ø dessine votre vaisseau spatial, et la ligne 14Ø permet de le déplacer.
Le compteur se trouve en ligne 15Ø.
La ligne 5ØØ provoque un beep-beep pendant toute la durée du jeu.

POUR JOUER :
Le jeu est lancé par l'ordre RUN.
Votre vaisseau apparaît vers le milieu de l'écran, et les météorites viennent à votre rencontre en partant du bas de l'écran. Pour vous déplacer, vous pouvez utiliser les touches de déplacement horizontal (→ et ←).
On peut arrêter la partie à tout moment en tapant ESC deux fois. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.

EXTENSIONS POSSIBLES :
Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.
On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

```
10 REM DANGER SPATIAL
20 RANDOMIZE(TIME)
30 EVERY 10 GOSUB 500
40 CLS
50 X=18:Z=0:Y=X
60 H=1+INT(RND*38)
70 IF RND<0.7 THEN LOCATE H,25:PEN 2:PRI
NT CHR$(231);
80 K=1+INT(RND*38)
90 IF RND<0.6 THEN LOCATE K,25:PEN 1:PRI
NT CHR$(231);
100 LOCATE 1,25:PRINT CHR$(10):B=TEST((X
-1)*16+8,(25-12)*16+8):C=TEST(X*16+8,(25
-12)*16+8)
110 IF B MOD 3 <>0 OR C MOD 3 <>0 THEN 1
90
120 LOCATE Y,11:PRINT STRING$(2,32);
130 LOCATE X,12:PEN 3:PRINT CHR$(242)+CH
R$(243);
140 A$=INKEY$
150 Y=X
160 X=X-(A$=CHR$(&F3) AND X<37)+(A$=CHR$
(&F2) AND X>1)
170 Z=Z+1
180 GOTO 60
190 TE=0:WHILE TE<200:SOUND 1,TE,1:TE=TE
+1:WEND
200 LOCATE 11,25:PAPER 3:PEN 1:PRINT "LE
 SCORE EST DE";Z:PAPER 0:PEN 1:END
500 SOUND 2,40,2:RETURN
```

Diago

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : difficile.

LE JEU :
On lance le programme par RUN et on obtient un premier affichage qui vous renseigne sur la performance à réaliser (par exemple, 5 points noirs ou 17 points noirs...), puis le temps est affiché et décompté, à partir de 1Ø secondes. Le jeu commence lorsque l'on arrive à Ø seconde. Des ronds apparaissent alors en bas et à gauche de l'écran, formant une ligne diagonale, et il vous faudra appuyer sur la touche Ø pour arrêter l'affichage des points dès que vous estimez que le nombre de points demandé au début est apparu à l'écran.
On obtient alors un affichage BRAVO ou PERDU selon le résultat obtenu.

LE PROGRAMME :
Le nombre de points noirs sera choisi au hasard, en ligne 5Ø.
Le temps est décompté par le sous-programme 24Ø-26Ø appelé toutes les secondes (ligne 3Ø).
Les points sont imprimés et la ligne diagonale est fabriquée par la ligne 12Ø.
La pression de la touche Ø est détectée en ligne 15Ø, et la ligne 17Ø contrôle la fin de partie (aucun réflexe).
Les affichages de fin sont exécutés par les lignes 17Ø, 2ØØ et 21Ø.

POUR JOUER :
On lancera l'exécution par RUN.
On obtient alors l'affichage de la phrase LE BON REFLEXE EST A... qui indique le nombre de points gagnants.
Le temps sera ensuite décompté sous la forme ATTENTION : IL RESTE ... SECONDES qui indique le temps restant avant l'apparition de la diagonale.
A l'affichage de Ø, la diagonale apparaît et commence à se déplacer sur l'écran.
Si l'on estime que le nombre de points réclamé a été atteint, on devra presser la touche Ø.
L'affichage indique alors BRAVO si l'on a gagné, ou PERDU dans le cas contraire ; puis la phrase AUTRE ESSAI (O/N) est affichée, répondre O pour faire une autre partie.

EXTENSIONS POSSIBLES :

On pourra envisager une adaptation du jeu à deux joueurs ou plus, en laissant la diagonale se former complètement et en testant la coïncidence lors de la pression de la touche affectée au joueur.

On peut aussi réaliser un graphisme plus soigné et modifier le principe du déplacement des objets sur l'écran (déplacement horizontal, vertical ou apparition au hasard...).

```
10 REM DIAGO
20 RANDOMIZE(TIME)
30 EVERY 50 GOSUB 240
40 CLS
50 X=INT(RND*15+3)
60 S=10
70 PRINT "LE BON REFLEXE EST A ";:PAPER
3:PRINT STRING$(X,202):PAPER 0
80 WHILE S>0:WEND
90 DI
100 CLS
110 U=1
120 LOCATE 1,26:PRINT TAB(U) CHR$(202)
130 FOR I=1 TO 100:NEXT I
140 U=U+1
150 A$=INKEY$
160 IF A$="0" THEN 190
170 IF U>22 THEN PEN 3:PRINT "PERDU!":WH
ILE TE<200:SOUND 2,TE,1:TE=TE+1:WEND:PEN
 1:GOTO 210
180 GOTO 120
190 IF U<>X+1 THEN U=23:GOTO 170
200 PEN 2:PRINT "BRAVO!":SOUND 1,478:SOU
ND 1,426:SOUND 1,379:PEN 1
210 PRINT:PRINT "UN AUTRE ESSAI (O/N)"
220 A$=INKEY$:IF A$="" THEN 220
230 IF UPPER$(A$)="O" THEN RUN ELSE END
240 LOCATE 5,12:PRINT USING "ATTENTION.
IL RESTE ## SECONDES";S
250 S=S-1:SOUND 1,200
260 RETURN
```

Œufs

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Un oiseau mystérieux se déplace en battant des ailes au hasard sur l'écran. Il pond des œufs (représentés par des O) que vous (un carré de couleur) devez essayer de récupérer. **Attention, seuls les œufs de votre couleur rapportent des points, les autres vous en enlèvent**. Votre score (nombre d'œufs récupérés) sera affiché en permanence, et la partie se déroule en un laps de temps défini.

LE PROGRAMME :
L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 7Ø et 11Ø ; son déplacement aléatoire est géré par le sous-programme 22Ø-31Ø (en particulier les lignes 27Ø à 3ØØ).
La ligne 13Ø permet de reboucler le programme sur la ligne 7Ø.
Le déplacement du joueur est régi par les lignes 15Ø et 16Ø, et le test permettant de détecter la présence d'un œuf est effectué en ligne 17Ø.
La ligne 19Ø permet l'affichage du score.
La fin de jeu est détectée en ligne 23Ø.

POUR JOUER :
On lancera le programme par RUN.
L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre ses œufs. Lorsque le symbole qui représente le joueur (un carré de couleur) apparaît sur l'écran, celui-ci peut se déplacer en utilisant les 4 touches de déplacement. Il vous faut dès lors essayer de récupérer un maximum d'œufs avant la fin de la partie.
On pourra arrêter de jouer à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
Sur le même principe, on pourra facilement imaginer d'autres thèmes de jeu. On pourra bien sûr améliorer la couleur, les sons, le graphisme, et éventuellement, définir un jeu à plusieurs joueurs, chacun essayant de récupérer un maximum d'œufs.

```
10 REM OEUFS
20 RANDOMIZE(TIME)
30 INK 0,9
40 SYMBOL AFTER 240:SYMBOL 240,24,24,126
,255,255,36,36,36
50 CLS:P=200:R=18:S=10:T=0
60 X=1+INT(RND*36):Y=1+INT(RND*21)
70 LOCATE X,Y:PEN 2:PRINT CHR$(198);CHR$
(240);CHR$(198);
80 FOR I=1 TO 30:A$=INKEY$
90 IF A$="" THEN NEXT I ELSE GOSUB 140
100 GOSUB 210
110 LOCATE X,Y:PEN 2:PRINT CHR$(193);CHR
$(240);CHR$(192);
120 GOSUB 210
130 GOTO 70
140 LOCATE R,S:PRINT " "
150 R=R-(A$=CHR$(&F3) AND R<38)+(A$=CHR$
(&F2) AND R>1)
160 S=S-(A$=CHR$(&F1) AND S<22)+(A$=CHR$
(&F0) AND S>1)
170 G=TEST ((R-1)*16+8,(25-S)*16+8)
180 T=T-2*(G=3)+(G=2)
190 LOCATE 15,23:PAPER 1:PEN 3:PRINT USI
NG " SCORE ###";T;:PAPER 0
200 RETURN
210 A=X:B=Y
220 P=P-1
230 IF P=0 THEN PRINT "C'EST FINI....":S
OUND 1,478:SOUND 1,426:SOUND 1,379:END
240 LOCATE R,S:PAPER 3:PEN 0:PRINT CHR$(
233);:PAPER 0
250 PEN 2-(RND<0.5):LOCATE A,B:PRINT " "
+CHR$(231)+" ";
260 X=ABS(X+INT(1+2*(RND<0.5)))
270 Y=ABS(Y+INT(1+2*(RND<0.5)))
280 IF X>37 OR X<1 THEN X=1
290 IF Y>22 OR Y<1 THEN Y=1
300 RETURN
```

Taquin de lettres

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Qui d'entre nous n'a pas, un jour ou l'autre, eu dans les mains un de ces petits puzzles où l'on essaie de déplacer d'une case à une autre des petits carrés de plastique colorés, jusqu'à l'obtention d'un résultat bien précis ? En son temps le 14-15 de SAM LOYD, a eu autant de succès que notre célèbre RUBIK'S.
Le taquin qui vous est proposé ici utilise 35 lettres, disposées au hasard dans un carré de 6 cases sur 6 cases, et ces lettres peuvent, replacées dans leur ordre initial, vous permettre de reconstituer un proverbe très connu. Une case coloriée vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME :
La grille de jeu est dessinée pour les boucles imbriquées K (7Ø-1ØØ) et I (4Ø-12Ø).
La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable R$ en ligne 13Ø.
Le mélange des lettres et l'impression sont faits par la boucle I (14Ø-22Ø).
La ligne 25Ø réalise l'affichage du carré coloré ; celui-ci est déplacé par les lignes 29Ø et 3ØØ en utilisant les 4 flèches.
La ligne 32Ø réimprime la lettre concernée à l'emplacement du carré coloré.
La ligne 34Ø comptabilise le nombre d'essais qui est imprimé grâce à la ligne 24Ø.

POUR JOUER :
On lance le programme par RUN.
La partie sera évidemment d'autant plus difficile que l'on ne connaît pas le proverbe à reconstituer. On peut déplacer les lettres et les faire passer à l'emplacement du carré coloré, à l'aide des 4 touches de direction (→, ←, ↓, ↑).
Le nombre d'essais est affiché en permanence, en dessous du puzzle.
On peut arrêter à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
Le même principe peut être appliqué à toute sorte de taquins (manipulation de nombres, de couleurs, de sons éventuellement). On peut aussi prévoir plusieurs phrases, avec un choix effectué par un générateur aléatoire, etc.

```
10 REM TAQUIN DE LETTRES
20 RANDOMIZE(TIME)
30 MODE 0
40 FOR I=1 TO 13
50 V=-I MOD 2:LOCATE 4,I+3
60 PRINT  CHR$(149-(2*V)+(I=1)+4*(I=13))
;
70 FOR K=1 TO 6
80 PRINT CHR$(154+122*NOT V);
90 PRINT CHR$(159+(10*NOT V)+(I=1)+4*(I=
13)+2*(K=6 AND V));
100 NEXT K
110 PRINT
120 NEXT I
130 A$="":R$="ONASOUVENTBESOIND'UNPLUSPE
TITQUESOI"
140 FOR I=1 TO 35
150 X=INT(RND*35+1)
160 IF MID$(R$,X,1)="-" THEN 150
170 A$=A$+MID$(R$,X,1)
180 MID$(R$,X,1)="-"
190 X=I MOD 6:Y=INT(I/6)
200 A$(X,Y)=MID$(A$,I,1)
210 LOCATE 2*X+5,2*Y+5:PEN 3:PRINT A$(X,
Y)
220 NEXT I
230 X=0:Y=0:S=0:C=32
240 LOCATE 1,21:PEN 6:PRINT USING "NOMBR
E D'ESSAIS :###";S
250 LOCATE 2*X+5,2*Y+5:PEN 3:PRINT CHR$(
143)
260 T$=INKEY$
270 IF T$="" THEN LOCATE 2*X+5,2*Y+5:PRI
NT CHR$(32):GOTO 250
280 I=X:J=Y
290 X=X-(T$=CHR$(&F3) AND X<5)+(T$=CHR$(
&F2) AND X>0)
300 Y=Y-(T$=CHR$(&F1) AND Y<5)+(T$=CHR$(
&F0) AND Y>0)
310 A$(I,J)=A$(X,Y):A$(X,Y)=""
320 LOCATE 2*I+5,2*J+5:PRINT A$(I,J)
330 IF I=X AND J=Y THEN PRINT CHR$(7);:G
OTO 250
340 S=S+1:SOUND 1,478:SOUND 1,379::SOUND
 1,319
350 GOTO 240
```

Cibles

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard.
Difficulté du jeu : facile.

LE JEU :
Il consiste simplement à tirer sur des cibles, en essayant de marquer le maximum de points ; le nombre de cibles (de 1 à 6) sera choisi en début de partie, et le score réalisé sera affiché en permanence. Si l'on touche le bord d'une cible, la partie s'arrête. On marquera d'autant plus de points que le nombre sur lequel on tombe sera plus grand.

LE PROGRAMME :
Le nombre de cibles est entré en ligne 5Ø. La boucle I (8Ø-1ØØ) effectue l'impression des cibles.
La ligne 12Ø affichera le score en permanence.
La balle est affichée par la ligne 13Ø.
La boucle I (15Ø-2ØØ) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 19Ø).
La ligne 22Ø regarde si l'on a tiré, et la ligne 23Ø regarde si la boule est encore sur la ligne de départ.
La ligne 24Ø fait progresser la boule vers la droite, et la ligne 25Ø vérifie si l'on a ou non atteint les cibles ; puis les lignes 26Ø et 27Ø regardent ce que l'on a touché.
La ligne 28Ø incrémente le score.
Les lignes 29Ø et 3ØØ s'occupent de la fin de partie.

POUR JOUER :
Lancer le programme par RUN, et taper le nombre de cibles désiré suivi de ENTER. Les cibles s'affichent, et la partie commence. Votre balle, représentée par un O, est affichée en bas à gauche ; vous pouvez la déplacer avec les flèches haut et bas (touches ↑ et ↓), et vous pouvez tirer à tout moment en tapant COPY.
Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît. Il est possible d'arrêter à tout moment en tapant ESC deux fois, sinon le jeu s'arrête au bout d'un certain nombre de tirs.

EXTENSIONS POSSIBLES :
On pourra sans difficulté augmenter le nombre de cibles, modifier la règle d'apparition des nombres dans les cibles pour la rendre moins hasardeuse, s'arranger pour que la balle elle-même soit représentée par un chiffre qui devrait coïncider à l'arrivée avec celui de la cible.

```
10 REM CIBLES
20 RANDOMIZE(TIME)
30 INK 0,9
40 PRINT:INPUT "NOMBRE DE CIBLES :";N
50 IF N<1 OR N>6 THEN 40
60 P=10:J=30:S=0:B=1
70 CLS
80 FOR I=1 TO N
90 LOCATE J+1,3*I:PRINT CHR$(140);CHR$(1
32):LOCATE J+1,3*I+1:PRINT " ";CHR$(133)
:LOCATE J+1,3*I+2:PRINT CHR$(131);CHR$(1
29)
100 NEXT I
110 K=0:A=18*16
120 LOCATE 8,21:PAPER 1:PEN 3:PRINT USIN
G " SCORE ### - BALLE ### ";S;B:PAPER 0
130 TAG: MOVE K,A:PRINT CHR$(231);:TAGOF
F
140 D=A
150 FOR I=1 TO N
160 U=INT(RND*P)
170 LOCATE J+1,3*I+1:PAPER 2:PRINT CHR$(
U+48):V(I)=U
180 A$=INKEY$
190 A=A+4*(A$=CHR$(&F1) AND A>6*16)-4*(A
$=CHR$(&F0) AND A<400)
200 NEXT I
210 TAG:MOVE K,D:PRINT " ";:TAGOFF
220 IF A$=CHR$(&E0) AND K=0 THEN B=B+1:G
OTO 240
230 IF K=0 THEN 130
240 K=K+16
250 IF K<16*J THEN 130
260 C=TEST(K+15,A-8)
270 IF C<>2 THEN LOCATE 12,12:PRINT "PER
DU!":WHILE TE<200:SOUND 1,TE,1:TE=TE+1:W
END:PAPER 0:END
280 S=S+V(ROUND((25-((A-8)/16))/3))
290 P=P-1:PRINT CHR$(7);
300 IF P=0 THEN SOUND 1,478:SOUND 1,426:
SOUND 1,379:END
310 GOTO 110
```

Gouttes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
C'est un véritable programme de dessin animé que l'on peut considérer, en fait, comme un interlude ; un récipient est dessiné sur l'écran, vu en coupe, et il contient déjà des obstacles. Une goutte d'eau, symbolisée par un rond bleu, apparaît au-dessus du récipient et tombe en remplissant le récipient au fur et à mesure. Sa trajectoire, lorsqu'elle rencontre un obstacle, est entièrement déterminée par le hasard : elle part à droite ou à gauche, jusqu'à ce qu'il n'y ait plus aucune possibilité. Alors, une autre goutte tombe, et ainsi de suite, jusqu'au débordement final.

LE PROGRAMME :
Les deux boucles emboîtées I (6Ø-13Ø) et J (7Ø-12Ø) dessinent le récipient et les obstacles.
La ligne 17Ø dessine la goutte et la ligne 21Ø envoie dans un sous-programme (37Ø-39Ø) qui permet de déterminer s'il y a ou non rencontre avec un obstacle.
Si c'est non, on va en 29Ø, et la goutte continue de tomber.
Si c'est oui, la goutte part à droite ou à gauche (ligne 23Ø) selon la valeur définie pour K en ligne 15Ø.
On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 24Ø, 25Ø ou 26Ø).
Selon le résultat du test, on utilisera 27Ø, 29Ø, 31Ø ou 34Ø.

POUR JOUER :
Le RUN déclenche le dessin du récipient et des obstacles, la goutte apparaît, se met à tomber et finit par se trouver une place. Une autre goutte apparaît et l'on recommence.
Le programme s'arrête de lui-même lorsque le récipient déborde ; on peut l'arrêter à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
On peut, bien entendu, ajouter la couleur et le son et aussi envisager un jeu dans lequel un petit personnage devrait échapper aux gouttes ou à la noyade.

```
10 REM GOUTTES
20 RANDOMIZE(TIME)
30 SYMBOL AFTER 240:SYMBOL 240,28,62,127
,127,127,62,28
40 EVERY 40 GOSUB 1000:DI
50 CLS
60 FOR I=12 TO 20
70 FOR J=12 TO 27
80 LOCATE 11,I:PEN 1:PRINT CHR$(143):LOC
ATE 28,I:PRINT CHR$(143)
90 LOCATE J,20:PRINT CHR$(143)
100 IF I<>12 AND I<>20 THEN LOCATE J,I:P
EN 3:PRINT CHR$(143):PEN 1
110 IF I<>20 AND RND<0.7 THEN LOCATE INT
(RND*15+12),I:PRINT " "
120 NEXT J
130 NEXT I
140 Y=INT(RND*7+18):X=6
150 K=-2*(RND<0.5)-1
160 I=0:EI
170 LOCATE Y,X:PEN 2:PRINT CHR$(240)
180 FOR J=0 TO 100:NEXT J
190 LOCATE Y,X:PRINT " "
200 U=X+1:V=Y
210 GOSUB 370
220 IF J=0 THEN 290
230 U=X:V=Y+K
240 GOSUB 370
250 IF J=0 THEN 310
260 IF I=0 THEN 340
270 LOCATE Y,X:PRINT CHR$(240)
280 GOTO 140
290 X=X+1
300 GOTO 150
310 Y=Y+K
320 I=1
330 GOTO 170
340 K=-K
350 I=1
360 GOTO 230
370 J=TEST((V-1)*16+8,(25-U)*16+8)
380 IF U=24 THEN LOCATE 14,6:PRINT "C'ES
T FINI...":SOUND 1,478:SOUND 1,379:SOUND
 1,319:END
390 RETURN
1000 SOUND 129,0:SOUND 1,20,5:RETURN
```

Obstacles

REFERENCES :
Difficulté de programmation : assez difficile.
Catégorie de jeu : jeu de réflexe (pour un joueur).
Difficulté du jeu : moyenne.

LE JEU :
Des obstacles se déplacent sur l'écran pour venir à la rencontre d'une petite voiture qui dispose de la possibilité de se déplacer vers la gauche ou vers la droite pour les éviter et qui devra tenir le plus longtemps possible.
La distance parcourue sera affichée en permanence, et le jeu se termine lorsque la voiture rencontre, soit le bord du terrain, soit un obstacle.

LE PROGRAMME :
La ligne 8Ø fait avancer les obstacles vers la voiture du joueur qui est imprimée grâce à la ligne 7Ø.
Les lignes 8Ø, 9Ø, 1ØØ fabriquent la route (les lignes 9Ø et 1ØØ bien qu'identiques sont toutes deux indispensables, car la valeur de RND ne sera, en principe, pas la même sur ces deux lignes).
La ligne 12Ø affiche le score dans une fenêtre # 1 protégée contre le défilement de la route.
La ligne 14Ø permettra de déplacer le véhicule.
Les lignes 15Ø et 16Ø permettent de contrôler s'il y a ou non un obstacle et de continuer le programme s'il y a lieu.
Le score est incrémenté en ligne 16Ø.
En ligne 17Ø, l'utilisation d'une encre transparente permet de montrer la voiture écrasant l'obstacle.

POUR JOUER :
On lance le programme en tapant RUN.
La route et les obstacles apparaissent en bas de l'écran et montent vers la voiture. La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches → et ←).
La rencontre avec un obstacle ou la bordure interrompt le déroulement du programme et provoque l'affichage du score.
On peut, cependant, sortir du programme à tout moment en utilisant la touche ESC deux fois.

EXTENSIONS POSSIBLES :
Il sera assez facile d'améliorer le dessin de la voiture, de la route, des obstacles. On pourra améliorer le son et la couleur, admettre un certain nombre d'accidents, enregistrer le meilleur score (pour plusieurs joueurs), éventuellement dessiner plusieurs voitures, etc.

```
10 REM OBSTACLES
20 RANDOMIZE(TIME)
30 EVERY 5 GOSUB 1000
40 SYMBOL AFTER 240:SYMBOL 240,93,127,85
,20,93,127,93,8:SYMBOL 241,8,20,8,62,42,
8,28,54
50 CLS:WINDOW #0,1,11,1,25
60 S=0:Y=5
70 LOCATE Y,1:PEN 3:PRINT CHR$(240)
80 LOCATE 1,25:PEN 1:PRINT CHR$(143);STR
ING$(8,32);CHR$(143)
90 IF RND<0.7 THEN LOCATE INT(RND*8+2),2
5:PEN 2:PRINT CHR$(241);
100 IF RND<0.8 THEN LOCATE INT(RND*8+2),
25:PEN 3:PRINT CHR$(241)+CHR$(10)
120 PEN #1,1+(S MOD 3):LOCATE #1,15,15:P
RINT #1, USING "SCORE :###";S
130 A$=INKEY$
140 Y=Y-(A$=CHR$(&F3) AND Y<10)+(A$=CHR$
(&F2) AND Y>1)
150 C=TEST((Y-1)*16+8,(25-1)*16+8)
160 IF C=0 THEN S=S+1:GOTO 70
170 LOCATE Y,1:PRINT CHR$(22);CHR$(1);CH
R$(240)
180 WHILE TE<200:SOUND 1,TE,1:TE=TE+1:WE
ND
190 END
1000 SOUND 1,400,2,,,2:RETURN
```

Captures dans l'espace

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : moyenne.

LE JEU :
Votre vaisseau spatial navigue dans un dangereux champ de météorites et votre mission est de vous emparer de la plus grande quantité possible de minerai d'urane, un métal indispensable que l'on ne peut trouver que dans ces lieux particulièrement dangereux.
Vous devez arriver à manœuvrer pour recueillir l'urane, sans vous faire heurter par une météorite.

LE PROGRAMME :
La ligne 7Ø permet de faire apparaître le minerai d'urane.
La ligne 9Ø fera apparaître les météorites et en assure le déplacement sur l'écran.
Les lignes11Ø à 13Ø contrôlent le déroulement du jeu (a-t-on heurté une météorite ou a-t-on capturé un bloc d'urane ?).
La rencontre d'une météorite provoque la fin de partie.
Votre vaisseau spatial est imprimé par la ligne 15Ø et son déplacement est contrôlé par la ligne 16Ø.

POUR JOUER :
On lance le programme par RUN et les météorites commencent à monter sur l'écran, ainsi que les blocs d'urane.
Les touches de déplacement horizontal (→ et ←) seront utilisées pour déplacer votre vaisseau et récupérer l'urane, mais le moindre heurt avec une météorite provoquera la fin de la partie et l'affichage de la quantité d'urane capturée.
On pourra aussi arrêter à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
On peut rendre le jeu de plus en plus difficile en augmentant progressivement le nombre de météorites, ou encore, en minutant la durée de la partie.
On peut aussi prévoir un affichage permanent du score, introduire un deuxième vaisseau spatial manœuvré par une puissance étrangère et qui cherche aussi à s'emparer de l'urane...

```
10 REM CAPTURES DANS L'ESPACE
20 RANDOMIZE(TIME)
30 INK 1,0
40 CLS:SYMBOL AFTER 240:SYMBOL 240,0,8,8
,54,127,28,8
50 X=15:Z=0
60 S=INT(1+RND*39)
70 LOCATE S,25:PEN 2:PRINT "#";
80 S=INT(1+RND*39)
90 IF RND<0.7 THEN LOCATE S,25:PEN 3:PRI
NT "*";:PRINT CHR$(10)
100 FOR I=1 TO 10:NEXT I
110 C=TEST((X-1)*16+8,(25-12)*16+8)
120 IF C=3 THEN PAPER 3:PEN 1:LOCATE 11,
24:PRINT USING "SCORE OBTENU : ###";Z:WH
ILE TE<200:INK 0,TE MOD 26:SOUND 1,TE,1:
TE=TE+1:WEND:PAPER 0:END
130 IF C=2 THEN Z=Z+1:SOUND 1,478:SOUND
1,426:SOUND 1,379
140 LOCATE X-1,11:PRINT STRING$(3,32)
150 PEN 1:LOCATE X-1,12:PRINT " "+CHR$(2
40)+" "
160 A$=INKEY$
170 X=X-(A$=CHR$(&F3) AND X<38)+(A$=CHR$
(&F2) AND X>2)
180 GOTO 60
```

Anomalie

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Il repose essentiellement sur une bonne capacité d'observation. Deux grilles de 25 cases chacune sont dessinées sur l'écran. Des symboles identiques sont affichés en des endroits différents des deux grilles, mais l'un des symboles a été modifié sur la grille de droite. Le jeu consiste à trouver lequel, en entrant ses coordonnées dans l'ordre ligne-colonne.

LE PROGRAMME :
La boucle I (5Ø-8Ø) imprime les chiffres de repérage des grilles et des points, avec une encre masquée.
La boucle principale I (1ØØ-16Ø) imprime les symboles des deux grilles, dans un ordre différent.
La ligne 11Ø choisit le caractère à imprimer.
Le sous-programme (1ØØØ-1Ø4Ø) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles). Lors du tirage de la lettre modifiée, la ligne 1Ø3Ø s'assure que celle-ci n'est pas identique à celle qu'elle remplace. La matrice B contient la valeur de chaque lettre de la grille de droite.
Les lignes 17Ø et 18Ø changent, de façon aléatoire, l'un des caractères.
Les coordonnées du point essayé sont entrées aux lignes 19Ø et 2ØØ, et le contrôle est effectué par la ligne 21Ø.
La ligne 22Ø s'occupe de l'affichage du résultat.

POUR JOUER :
Après avoir lancé le programme par l'instruction RUN, on obtient l'affichage des deux grilles. On devra alors chercher, par comparaison, quel est le caractère qui a été modifié (sur la grille de droite).
On entre alors son numéro de ligne, suivi de ENTER, et son numéro de colonne, suivi de ENTER.
Le programme répondra alors BRAVO ou PERDU.

EXTENSIONS POSSIBLES :
Il sera possible d'utiliser des symboles graphiques de diverses couleurs, ce qui pourra simplifier la recherche.
On peut aussi envisager d'utiliser des dessins plus sophistiqués, améliorer le contrôle du résultat, admettre la bonne réponse sur la grille de droite ou celle de gauche, introduire un minutage, etc.

```
10 REM ANOMALIE
20 INK 1,1:INK 2,1:INK 3,1:RANDOMIZE(TIM
E)
30 FOR I=1 TO 5:PO$=PO$+CHR$(144)+" ":NE
XT I
40 T=5:DIM B(5,5):CLS
50 FOR I=1 TO 5
60 LOCATE 2*I+1,1:PRINT I:LOCATE 1,2*I+1
:PRINT I
70 LOCATE 2*I+19,1:PRINT I:LOCATE 19,2*I
+1:PRINT I
75 LOCATE 4,2*I+1:PRINT PO$:LOCATE 22,2*
I+1:PRINT PO$
80 NEXT I
100 FOR I=1 TO 22
110 D=INT(26*RND+1)
120 L=0:S=T
130 GOSUB 1000
140 L=18
150 GOSUB 1000
160 NEXT I
170 S=0
180 GOSUB 1000:INK 1,24:INK 2,6:INK 3,9
190 PEN 1:LOCATE 1,16:INPUT "LIGNE......
.";I
200 PRINT:INPUT "COLONNE ....";T
210 LOCATE 1,20:IF B(T,I)=D THEN PRINT "
BRAVO!...." ELSE PRINT "PERDU!....":WHIL
E TE<200:SOUND 1,TE,1:TE=TE+1:WEND
220 PRINT:PRINT "LA LETTRE A TROUVER ETA
IT ";CHR$(64+D):END
1000 U= 3+2*INT(RND*T):V=4+L+2*INT(RND*S
)
1010 C=TEST((V-1)*16+8,(25-U)*16+8)
1020 IF C<>1 AND S<>0 THEN 1000
1030 IF S=0 THEN IF B((V-20)/2,(U-1)/2)=
D THEN 1000
1040 IF L<>0 THEN B((V-20)/2,(U-1)/2)=D
1050 LOCATE V,U:PEN 2-(L<>0):PRINT CHR$(
D+64)
1060 RETURN
```

Plusieurs en un

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : variable, selon l'utilisation.

LE JEU :
Plusieurs jeux sont en réalité possibles avec ce programme qui va se comporter comme un crayon électronique vous permettant d'inscrire des symboles de couleur, en n'importe quel emplacement de votre feuille-écran de télévision. On pourra, par exemple, l'utiliser pour un jeu de morpion (le premier à réaliser un alignement de 5 pions identiques gagne) ou pour un GO (on réalise des encadrements des pions adverses) ou tout autre jeu, selon son imagination.

LE PROGRAMME :
Le terrain de jeu est délimité par un cadre mis en place par la ligne 3Ø.
Le changement de joueur est effectué par la ligne 5Ø.
Le joueur qui a le trait voit le symbole qu'il a choisi (carré de couleur) se placer en limite du cadre (ligne 7Ø).
Les lignes 1ØØ et 11Ø permettent aux deux joueurs de déplacer le symbole sur le terrain.
Les lignes 13Ø et 14Ø contrôlent si le déplacement est possible (case libre et non dépassement des limites).
La ligne 18Ø permet d'écrire sur une case libre.

POUR JOUER :
RUN provoque l'affichage des limites du terrain et du numéro du joueur, ainsi que des symboles correspondants que l'on devra déplacer sur l'écran à l'aide des 4 touches de direction (→, ←, ↓ , ↑).
Il n'est pas possible de venir sur une case déjà occupée et il faudra appuyer sur la touche COPY pour inscrire son propre symbole dans une case disponible.
Dès que cela a été fait, le symbole du second joueur apparaît dans la colonne de gauche, à peu près au milieu, et le joueur procédera de la même manière que son adversaire.
Pour sortir du programme, il suffira de taper ESC deux fois.

EXTENSIONS POSSIBLES :
On pourra utiliser l'écran tout entier comme terrain de jeu et réaliser des contrôles en fonction des différents jeux que l'on désire pratiquer : contrôle d'alignements de 5 pièces pour le morpion, contrôle du nombre de pièces capturées pour le GO.

```
10 REM PLUSIEURS EN UN
20 MODE 1
30 MOVE 8,8:DRAWR 0,24*16,1:DRAWR 25*16,
0,1:DRAWR 0,-24*16,1:DRAWR -25*16,0,1
40 J=0
50 J=ABS(J-1)
60 X=13:Y=1
70 LOCATE Y,X:PEN 2-(J=1):PRINT CHR$(231
-28*J)
80 A$=INKEY$:IF A$="" THEN 80
90 A=X:B=Y
100 X=X-(A$=CHR$(&F1) AND X<24)+(A$=CHR$
(&F0) AND X>2)
110 Y=Y-(A$=CHR$(&F3) AND Y<25)+(A$=CHR$
(&F2) AND Y>2)
120 LOCATE B,A:PRINT " "
130 C=TEST((Y-1)*16+8,(25-X)*16+8)
140 IF C<>2 AND C<>3 AND C<>1 THEN 160
150 X=A:Y=B
160 CA=INKEY(9)
170 IF CA<>0 OR Y=2 THEN 70
180 LOCATE Y,X:PEN 2-(J=1):PRINT CHR$(23
1-28*J)
190 IF J=0 THEN SOUND 1,200,10 ELSE SOUN
D 1,800,10
200 GOTO 50
```

Parcours du cavalier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il se présente, ici, comme un jeu de hasard total, mais provient, en réalité, d'un problème assez difficile à résoudre : comment déplacer un cavalier sur un échiquier de façon qu'il passe une fois, et une seule, sur chacune des 64 cases du damier.
Le cavalier se déplace suivant les règles communément admises aux échecs, et sa position de départ est définie par le joueur ; par contre, ses déplacements sont effectués au hasard. Arrivera-t-il à parcourir les 64 cases ?

LE PROGRAMME :
La ligne 6Ø définit les divers déplacements possibles, dans deux variables alphanumériques A$ et B$; la boùcle emboîtée I (7Ø-1ØØ) dessine le terrain de jeu.
On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 11Ø et 12Ø.
La ligne 14Ø imprime le numéro du coup.
La ligne 15Ø permet le tirage au hasard d'un déplacement du cavalier.
La boucle G (16Ø-21Ø) cherche si le déplacement est possible ; si ce n'est pas le cas, on essaie tous les autres (toujours dans le même ordre).
Les lignes 17Ø et 18Ø effectuent le déplacement.
La ligne 19Ø contrôle que l'on ne sorte pas du terrain de jeu.
La ligne 2ØØ envoie en ligne 23Ø si le coup est possible.
La ligne 22Ø gère la fin de partie (cavalier bloqué) et affiche le score.
Les lignes 23Ø, 24Ø et 25Ø réalisent le coup et incrémentent le score.

POUR JOUER :
On lance le programme par l'ordre RUN. Le terrain de jeu s'affiche ; on choisit alors la position de départ en tapant d'abord un numéro de ligne (entre 1 et 8), suivi de ENTER puis son numéro de colonne, suivi de ENTER. Le jeu commence immédiatement et il se déroule de façon entièrement automatique jusqu'à ce que le cavalier soit bloqué. On peut, bien sûr, arrêter à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
On peut imaginer de nombreuses variantes sur ce thème, ne serait-ce qu'en commençant par faire varier la taille des échiquiers ; on peut aussi améliorer considérablement le graphisme. Mais, le plus intéressant est d'arriver à définir un algorithme donnant la ou les solutions du problème initial...

```
10 REM PARCOURS DU CAVALIER
20 RANDOMIZE(TIME)
30 INK 0,9:BORDER 9
40 CLS
50 S=1:T=0
60 A$="56653223":B$="653223"+A$+"56":A$=
A$+A$
70 FOR I=1 TO 8
80 LOCATE 3*I+2,1:PRINT I
90 LOCATE 1,2*I+1:PRINT I;"  .  .  .  .
 .  .  .  ."
100 NEXT I
110 LOCATE 1,19:INPUT "LIGNE DE DEPART
:";I
120 PRINT:INPUT "COLONNE DE DEPART:";J
130 IF S>9 THEN S=0:T=T+1
140 LOCATE 3*J+3,2*I+1:PAPER 2:PEN 3:PRI
NT RIGHT$(STR$(S),1):IF T<>0 THEN LOCATE
 3*J+2,2*I+1:PRINT RIGHT$(STR$(T),1)
150 Z=INT(RND*8+1)
160 FOR G=Z TO Z+8
170 A=I+VAL(MID$(A$,G,1))-4
180 B=J+VAL(MID$(B$,G,1))-4
190 IF  A<1 OR B<1 OR A>8 OR B>8 THEN 21
0
200 IF TEST (((3*B+3)-1)*16+8,(25-(2*A+1
))*16+8)=0 THEN 230
210 NEXT G
220 PAPER 1:PEN 3:LOCATE 1,23:PRINT "TOT
AL :";T*10+S:SOUND 1,478:SOUND 1,426:SOU
ND 1,379:PAPER 0:PEN 1: END
230 I=A:J=B
240 S=S+1:PRINT CHR$(7);
250 GOTO 130
```

Othello

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

LE JEU :
Devenu un grand classique des jeux sur micro-ordinateur, le jeu d'Othello ou Reversi gagne, même si comme ici il est pratiqué à deux joueurs, à être géré sur ordinateur, avec visualisation : en effet, le jeu consistant essentiellement en des retournements des pions de l'adversaire, ceci peut se révéler, à la longue, très fastidieux, surtout si vous êtes un joueur assidu.
Le principe est que le joueur possédant les X doit pour pouvoir jouer, réussir à encadrer une ou plusieurs des pièces de son adversaire, dans une ou plusieurs directions ; s'il y réussit, les O seront alors retournés pour devenir des X (le retournement pourra être fait horizontalement, verticalement ou suivant les diagonales).
Le joueur ayant les O procédera de la même façon, et ceci jusqu'à ce que le damier soit rempli ou que la personne ne puisse plus jouer (si un joueur ne peut retourner des pions, il doit passer son tour).

LE PROGRAMME :
Les boucles emboîtées P (5Ø-12Ø) et Q (7Ø-11Ø) dessinent le terrain de jeu et la ligne 14Ø affiche un X ou un O, selon le joueur qui commence.
La boucle P (2ØØ-32Ø) contrôle la validité du coup (la case est-elle libre : lignes 3ØØ et 31Ø), en utilisant une partie de programme qui se trouve aux lignes 36Ø à 42Ø (a-t-on pris un ou plusieurs ennemis en ligne 36Ø et retournement éventuel autorisé par la ligne 38Ø : on refait un balayage dans toutes les directions, si l'encadrement s'est révélé possible).
La ligne 2ØØ définit les 8 directions d'exploration.
Les lignes 24Ø et 25Ø calculent les déplacements à effectuer pour explorer ces 8 directions.
La ligne 23Ø réalise les déplacements.
La ligne 34Ø assure le changement de joueur, s'il y a eu retournement (contrôlé en ligne 33Ø).

POUR JOUER :
Après RUN, chaque joueur entrera à son tour les coordonnées (ligne, ENTER, puis colonne, ENTER) de la case où il désire jouer ; si le coup est accepté, les pions sont retournés et l'on change de joueur ; si un coup n'est pas possible, on entre Ø en première coordonnée pour le changement de joueur.

EXTENSIONS POSSIBLES :

Avant tout, bien sûr, améliorer l'affichage avec des couleurs, une grille plus aérée.

On pourra aussi imaginer un jeu machine pour AMSTRAD.

Après avoir lancé le programme par RUN, il convient d'entrer la dimension de la grille (4, ENTER ou 6, 8 ...).

```
10 REM OTHELLO
15 WINDOW #1,25,40,8,18
20 INPUT "DIMENSION DE LA GRILLE :";A
30 IF A MOD 2 <>0 OR A>8 OR A<2 THEN 20
40 CLS
50 FOR P=1 TO A
60 LOCATE 2*P+1,1:PRINT RIGHT$(STR$(P),1
)
70 FOR Q=1 TO A
80 LOCATE 1,2*P+1:PRINT RIGHT$(STR$(P),1
):LOCATE 2*Q+1,2*P+1:PRINT CHR$(144)
90 IF Q=A/2+1 AND P=A/2 THEN LOCATE 2*Q-
1,2*P+1:PEN 2:PRINT CHR$(231);" ";:PEN 3
:PRINT CHR$(203):PEN 1
100 IF Q=A/2+1 AND P=A/2+1 THEN LOCATE 2
*Q-1,2*P+1:PEN 3:PRINT CHR$(203);" ";:PE
N 2:PRINT CHR$(231):PEN 1
110 NEXT Q
120 NEXT P
130 K=1
140 CLS #1:LOCATE #1,2,3:PEN #1,2-(K=1):
PRINT #1,"JOUEUR ";CHR$(231-K*28):SOUND
1,200+200*K,10
160 PRINT #1:INPUT #1,"LIGNE  :";U
170 IF U=0 THEN 340
180 PRINT #1:INPUT #1,"COLONNE:";V
190 H=1
200 FOR P=1 TO 8
210 S=0
220 W=0
230 T=2*U+1:R=2*V+1
240 F=(P>3 AND P<7)-(P<3 OR P>7)
250 G=(P>1 AND P<5)-(P>=6)
260 Q=-203*(S=1)-231*(S=2)-144*(S=0)
```

```
270 IF S=0 THEN 290
280 LOCATE R,T: PEN 2-(K=1):PRINT CHR$(Q
)
290 T=T+2*F:R=R+2*G
300 C=TEST((R-1)*16+8,(25-T)*16+8)
310 IF C=2 OR C=3 THEN 360
320 NEXT P
330 IF H<>0 THEN 140
340 K=ABS(K-1)
350 GOTO 140
360 IF (C=2 AND K=1) OR (C=3 AND K=0) TH
EN W=1
370 IF S<>0 THEN LOCATE R,T:PEN 2-(K=1):
PRINT CHR$(Q)
380 IF W=1 AND ((C=2 AND K=1) OR (C=3 AN
D K=0)) THEN 290
390 IF W=0 OR S<>0 THEN 320
400 S=1-(K=0)
410 H=-H*(S=0)
420 GOTO 220
```

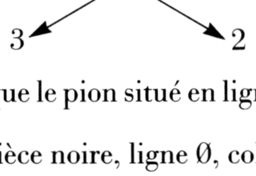
Dames

REFERENCES :
Difficulté de programmation : très difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
C'est le jeu classique de dames à la française, mais un peu incomplet puisqu'il ne prévoit pas la possibilité de faire des dames. La règle du jeu reste classique, mais les contrôles ont été un peu simplifiés. Les mouvements des pièces sont entrés en une seule manœuvre (position de départ et direction du déplacement) et les prises sont réalisées automatiquement, si elles sont possibles.
Par contre le changement de joueur étant réalisé à chaque coup, il faudra éventuellement taper Ø pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.

LE PROGRAMME :
Les boucles I (8Ø-17Ø) et J (9Ø-16Ø) dessinent le damier et définissent les positions initiales des pions noirs et des pions blancs. Les positions sont dans une table R (X, Y) et la couleur dans une autre table C (X, Y).
L'impression des pions proprement dite est effectuée par les boucles emboîtées J (19Ø-25Ø) et I (2ØØ-24Ø).
La ligne 27Ø indique le joueur qui a le trait.
Le coup est entré en ligne 27Ø.
La ligne 28Ø réalise le changement de joueur (on tape Ø).
La ligne 3ØØ regarde si le déplacement est possible.
La ligne 33Ø traite le cas d'un déplacement et la ligne 34Ø celui d'un saut.
La ligne 38Ø assure le changement de joueur.

POUR JOUER :
Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et un cercle de couleur apparaît en dessous pour indiquer que le trait appartient au joueur correspondant.
On joue en indiquant par un mot de 3 lettres (pouvant commencer par un Ø) la ligne et la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :



Par exemple 9D1 signifie que le pion situé en ligne 9, colonne D, se déplace dans la direction 1.
De même ØG3 signifiera pièce noire, ligne Ø, colonne G, direction 3.

Le changement de joueur étant effectué automatiquement il faudra taper Ø pour rejouer si le coup est refusé ou s'il y a eu prise. Le programme ne contrôle pas l'arrivée à dame, et n'interdit pas les déplacements en arrière.

EXTENSIONS POSSIBLES :

Il sera assez facile de commencer par améliorer les contrôles (changement de joueur, coup en arrière qui ne correspondrait pas à une prise, sortie de terrain). On pourra envisager la création de dames, leur déplacement, et éventuellement écrire un programme permettant à votre AMSTRAD de se mêler avec de bonnes chances de réussite à vos passionnantes parties.

```
10 REM DAMES
20 DIM R(10,10):DIM C(10,10):SYMBOL AFTE
R 240:SYMBOL 240,0,0,0,0,0,0,0:SYMBOL
241,0,60,195,195,189,129,66,60
30 A$=STRING$(3,140)
40 B$=STRING$(3,143)
50 C$=STRING$(3,131)
60 CLS
70 PRINT "     A  B  C  D  E  F  G  H  I
 J"
80 FOR I=1 TO 10
90 FOR J=1 TO 10
100 IF (I+J) MOD 2=0 THEN 120
110 LOCATE 3*J+1,2*I:PRINT A$:PRINT I-1:
LOCATE 3*J+1,2*I+1:PRINT B$:LOCATE 3*J+1
,2*I+2:PRINT C$
120 R(I,J)=0
130 IF (I+J) MOD 2=1 THEN 160
140 IF J<5 THEN R(I,J)=1:C(I,J)=2
150 IF J>6 THEN R(I,J)=1:C(I,J)=3
160 NEXT J
170 NEXT I
180 H=1
190 FOR J=1 TO 10
200 FOR I=1 TO 10
210 IF (I+J) MOD 2 <>0 THEN 240
220 X=3*J+2:Y=2*I+1
230 LOCATE X,Y:PEN C(I,J):PRINT CHR$(240
+R(I,J))
240 NEXT I
```

```
250 NEXT J
260 LOCATE 1,23:PRINT CHR$(20);
270 LOCATE 1,23:PEN 2-(H=1):PRINT "JOUEU
R ";CHR$(241);:INPUT " - VOTRE COUP :";D
$
280 IF D$="O" THEN H=ABS(H-1):GOTO 260
290 GOSUB 400
300 IF R(L,C)=0 OR (C(L,C)=2 AND H=1) OR
(C(L,C)=3 AND H=0) THEN 260
310 T=(1 AND (D=1 OR D=2)) - (1 AND (D=4
 OR D=3))
320 S=(1 AND (D=3 OR D=2)) - (1 AND (D=1
 OR D=4))
330 IF R(L+S,C+T)=0 THEN R(L,C)=0:R(L+S,
C+T)=1:C(L+S,C+T)=2-(H=1):GOTO 380
340 IF C(L+S,C+T)=2-(H=0) AND R(L+S,C+T)
=1 AND R(L+2*S,C+2*T)=0 THEN 360
350 GOTO 260
360 R(L,C)=0:R(L+S,C+T)=0:R(L+2*S,C+2*T)
=1:C(L+2*S,C+2*T)=2-(H=1)
370 GOTO 190
380 H=ABS(H-1)
390 GOTO 190
400 L=VAL(LEFT$(D$,1))+1:C=ASC(MID$(D$,2
,1))-64:D=VAL(RIGHT$(D$,1))
410 RETURN
```

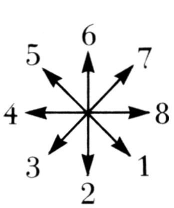
Quatre à quatre

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (un ou deux joueurs).
Difficulté du jeu : difficile.

LE JEU :
Il pourra être pratiqué à un ou plusieurs joueurs. Dans les deux cas, le but est d'arriver à échanger les pions X avec les pions O ; si l'on joue en solitaire, cet échange devra être réalisé avec le moins de coups possible. Les pions sont déplacés alternativement en respectant les règles de déplacement du cavalier aux échecs ; il n'y a pas de prise, et on ne peut passer son tour.

LE PROGRAMME :
La variable alphanumérique A$, en ligne 4Ø, contient la table des coups possibles (déplacement du cavalier).
La boucle I (6Ø-1ØØ) dessine le terrain de jeu, et les positions de départ.
La ligne 12Ø permet d'afficher O ou X, selon le joueur qui a le trait.
La ligne 15Ø permet l'affichage du curseur clignotant qui indique sur quelle case on se trouve.
Les lignes 18Ø et 19Ø permettent le déplacement du curseur.
La ligne 2ØØ regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.
La ligne 21Ø réimprime la figure qui était sous le curseur. La ligne 24Ø transforme en nombre le symbole chiffre frappé. La ligne 23Ø réalise le contrôle de départ.
Les lignes 25Ø et 26Ø effectuent le déplacement. La ligne 29Ø réalise le contrôle d'arrivée. Le résultat est imprimé par la ligne 3ØØ, le changement de joueur est réalisé par la ligne 31Ø et un "." est imprimé à l'emplacement de départ. Le sous-programme 33Ø-34Ø est utilisé pour les divers contrôles.

POUR JOUER :
On lance le programme par RUN. Le terrain de jeu est affiché, ainsi que le symbole permettant d'identifier le joueur qui a le trait. Le joueur peut choisir le pion qu'il désire déplacer en manœuvrant le curseur clignotant grâce aux touches de déplacement du curseur.
Pour déplacer son pion, il suffit, lorsque le curseur est sur ce pion, de taper la direction du déplacement, suivant le code :



Le coup sera refusé si la case de départ n'est pas conforme ou si la case d'arrivée est occupée. Sinon, le coup est réalisé, et le trait passe à l'autre joueur, qui déplace le curseur clignotant pour l'amener sur l'un des pions, et joue de la même façon.
Le joueur gagnant est celui qui réussit à amener le premier tous ses pions dans les cases initialement occupées par l'adversaire. On peut interrompre la partie à tout moment en tapant ESC deux fois.

EXTENSIONS POSSIBLES :
On peut améliorer les contrôles, prévoir un test de fin, utiliser un terrain de jeu différent ; on pourra aussi envisager l'écriture d'un programme permettant à votre AMSTRAD de participer au jeu avec de bonnes chances de l'emporter.

```
10 REM QUATRE A QUATRE
20 INK 0,15:BORDER 15
30 MODE 0
40 H=1:A$="5665322356"
50 LOCATE 5,2:PRINT "1 2 3 4 5"
60 FOR I=1 TO 5
70 LOCATE 1,3*I+3:PRINT RIGHT$(STR$(I),1
):LOCATE 5,3*I+3:PRINT CHR$(144)+" "+CHR
$(144)+" "+CHR$(144)+" "+CHR$(144)+" "+C
HR$(144)
80 IF I<3 THEN PEN 2:LOCATE 5,3*I+3:PRIN
T CHR$(231);" ";CHR$(231):PEN 1
90 IF I>3 THEN PEN 3:LOCATE 11,3*I+3:PRI
NT CHR$(203);" ";CHR$(203):PEN 1
100 NEXT I
110 X=1:Y=1
120 LOCATE 5,22:PEN 2+H:PRINT "JOUEUR  "
;CHR$(231-28*H)
130 GOSUB 330
140 I=J
150 LOCATE 3+2*X,3+3*Y:PRINT " "
160 A=X:B=Y
170 R$=INKEY$
180 X=X-(R$=CHR$(&F3) AND X<5)+(R$=CHR$(
&F2) AND X>1)
190 Y=Y-(R$=CHR$(&F1) AND Y<5)+(R$=CHR$(
&F0) AND Y>1)
200 IF R$<"9" AND R$>"0" THEN 230
```

```
210 LOCATE 3+2*A,3+3*B:PEN J:IF J=1 THEN
 PRINT CHR$(144) ELSE PRINT CHR$(231-28*
(J-2))
220 GOTO 120
230 IF (I=1) OR (I=2 AND H=1) OR (I=3 AN
D H=0) THEN GOTO 210
240 K=VAL(R$)
250 S=Y:Y=Y+VAL(MID$(A$,K,1))-4
260 T=X:X=X+VAL(MID$(A$,K+2,1))-4
270 IF X<1 OR X>5 OR Y<1 OR Y>5 THEN Y=S
:X=T:A=T:B=S:GOTO 210
280 GOSUB 330
290 IF J<>1 THEN X=-1:J=2+H:GOTO 270
300 LOCATE 3+2*X,3*Y+3:PEN (2+H):PRINT C
HR$(231-28*H)
310 H=ABS(H-1):I=1
320 GOTO 210
330 J=TEST ((3+2*X-1)*32+16,(25-(3+3*Y))
*16+8)
340 RETURN
```

Echecs

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexion, pour deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
Le programme propose seulement à deux joueurs de disputer une partie en se servant de l'écran du téléviseur comme plateau de jeu. Il n'y a pas de contrôle de la validité des coups, mais seulement gestion de la position des pièces sur l'échiquier. Les pions et les pièces sont dessinés à l'aide de l'instruction SYMBOL. La couleur permet de distinguer chaque camp.

LE PROGRAMME :
Les lignes 110 à 200 dessinent l'échiquier. Les lignes 150, 160, 170, 180, 210 et 220 définissent les conditions initiales, à l'aide de deux tableaux : R (I, J) pour les codes des pièces, et C (I, J) pour la couleur.
Les boucles emboîtées I (250-280) et J (240-290) dessinent les pièces.
La ligne 320 permet l'entrée de la ligne et de la colonne de départ, avec contrôle en ligne 330.
La ligne 360 permet d'entrer colonne et ligne d'arrivée.
La ligne 380 note la nouvelle position.
La ligne 390 assure le changement de joueur.

POUR JOUER :
On lance l'exécution par RUN.
L'échiquier est dessiné, puis les pièces, et enfin un texte de couleur indique le joueur qui a le trait ; on entre la colonne et la ligne de départ, sous la forme E 7 par exemple, on donne la colonne et la ligne d'arrivée. Le déplacement est alors effectué, et c'est au tour de l'autre joueur.
Pour arrêter, il suffit de taper ESC deux fois.

EXTENSIONS POSSIBLES :
On pourra chercher à réaliser un graphisme plus précis, introduire le contrôle du mouvement des pièces. Et pourquoi pas, écrire un programme qui permette à votre AMSTRAD de se comporter en partenaire honorable. Bon courage !

```
10 REM ECHECS
20 DIM R(8,8):DIM C(8,8):SYMBOL AFTER 24
0
30 SYMBOL 240,0,16,40,68,108,56,124,0:SY
MBOL 241,0,16,56,16,56,68,56,0
40 SYMBOL 242,0,84,124,56,56,124,124,0:S
YMBOL 243,0,84,40,16,108,124,124,0
50 SYMBOL 244,0,0,16,56,56,16,124,0:SYMB
OL 245,0,16,56,120,24,56,124,0:SYMBOL 24
6,0,0,0,0,0,0,0,0
60 A$=STRING$(3,140)
70 B$=STRING$(3,143)
80 C$=STRING$(3,131)
90 CLS:INK 1,9:INK 3,0
100 PRINT "    A  B  C  D  E  F  G  H"
110 FOR I=1 TO 8
120 FOR J=1 TO 8
130 IF (I+J) MOD 2=0 THEN 150
140 LOCATE 3*J,2*I:PRINT A$:PRINT CHR$(4
8+I):LOCATE 3*J,2*I+1:PRINT B$::LOCATE 3
*J,2*I+2:PRINT C$
150 IF I=2 OR I=7 THEN R(I,J)=4
160 IF I<>2 AND I<>7 THEN R(I,J)=6
170 C(I,J)=2
180 IF I=7 OR I=8 THEN C(I,J)=3
190 NEXT J
200 NEXT I
210 R(1,1)=2:R(1,2)=5:R(1,3)=0:R(1,4)=3:
R(1,5)=1:R(1,6)=0:R(1,7)=5:R(1,8)=2
220 FOR I=1 TO 8:R(8,I)=R(1,I):NEXT I
230 H=0
240 FOR J=1 TO 8
250 FOR I=1 TO 8
260 X=1+3*J:Y=2*I+1
270 LOCATE X,Y:PEN C(I,J):PAPER (I+J) MO
D 2:PRINT CHR$(240+R(I,J))
280 NEXT I
290 NEXT J
300 WINDOW #1,28,40,9,18
310 CLS #1:PEN #1,2+H:PRINT #1," JOUEUR
";:IF H=0 THEN PRINT #1,"BLEU" ELSE PRIN
T #1,"NOIR"
```

```
320 PRINT #1,"DEPART  (C,L)":LOCATE #1,7
,3:INPUT #1,"",D$
330 GOSUB 410
340 IF R(L,C)=6 OR C(L,C)=3-H THEN 310
350 T=R(L,C):S=C(L,C):R(L,C)=6
360 LOCATE #1,1,7:PRINT #1,"ARRIVEE (C,L
)":LOCATE #1,7,8:INPUT #1,"",D$
370 GOSUB 410
380 R(L,C)=T:C(L,C)=S
390 H=ABS(H-1)
400 GOTO 240
410 L=VAL(RIGHT$(D$,1)):C=ASC(LEFT$(D$,1
))-64
420 RETURN
```

Mur de briques

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

LE JEU :
S'il est un jeu que l'on peut présenter comme un grand classique des jeux d'arcades pour micro-ordinateur, c'est bien le mur de briques (en anglais BREAKOUT).
Il est offert, ici, dans une version simplifiée, mais offre néanmoins de bonnes performances : affichage du score en fin de partie, rebonds aléatoires, etc.
La règle en est très élémentaire : vous disposez d'une raquette qui peut être déplacée horizontalement pour renvoyer une balle qui va démolir les briques dessinées en haut de l'écran.
Votre problème est d'en démolir le plus grand nombre possible, sans rater la balle, ce qui provoquerait la fin de partie et l'affichage du score.

LE PROGRAMME :
La boucle I (5Ø-8Ø) dessine le terrain de jeu et les briques sont mises en place par les boucles emboîtées I (9Ø-13Ø) et J (1ØØ-12Ø).
La ligne 2ØØ dessine la raquette dont la position est contrôlée par la ligne 18Ø.
Les lignes 25Ø à 27Ø regardent si l'on a touché le mur, les briques ou la raquette.
Les rebonds sont gérés en conséquence avec des éventuels aléas, par les lignes 25Ø et 27Ø.
La ligne 21Ø affiche le score en permanence.

POUR JOUER :
Le programme sera lancé par l'instruction RUN, ce qui provoque l'affichage du terrain de jeu puis, très rapidement, l'envoi de la balle. Le joueur devra alors essayer de déplacer sa raquette à l'aide des touches de déplacement horizontal (→ et ←).
Si, à un moment quelconque, sa balle ne touche pas la raquette, le jeu se termine.
On peut aussi terminer à tout moment par ESC deux fois.

EXTENSIONS POSSIBLES :
On peut envisager d'agrandir le terrain de jeu, d'augmenter le nombre de briques, de prévoir plusieurs balles, d'introduire des effets sonores.
On peut aussi, en s'inspirant des jeux d'arcades, introduire de petits ouvriers qui viendront reconstruire votre mur de briques, au fur et à mesure que vous le démolissez.

```
10 REM MUR DE BRIQUES
20 DEF FN IN(X)=16*ROUND(X/16)
30 RANDOMIZE(TIME)
40 CLS:PEN 1:A$=CHR$(143):B$=CHR$(233)
50 FOR I=1 TO 18
60 LOCATE 10+I,1:PRINT A$
70 LOCATE 11,I+1:PRINT A$:LOCATE 28,I+1:
PRINT A$
80 NEXT I
90 FOR I=1 TO 6
100 FOR J=11 TO 26
110 LOCATE J+1,I+1:PEN 2+(I+J) MOD 2:PRI
NT B$
120 NEXT J
130 NEXT I
140 TAG
150 ORIGIN 11*16,7*16-1:I=0:J=8*16:K=16:
H=16:Y=6*16
160 A=I:B=J
170 A$=INKEY$
180 Y=Y-16*(A$=CHR$(&F3) AND Y<15*16)+16
*(A$=CHR$(&F2) AND Y>-4*16)
190 IF T=1 THEN MOVE IJ,II:PRINT " ";:T=
0:I=A:J=B
200 MOVE Y,-16:PRINT " "+STRING$(3,143)+
" ";
210 LOCATE #1,1,22:PRINT #1,"SCORE:";S
220 A=I+H:B=J+K:IA=FN IN(A):IB=FN IN(B):
II=FN IN(I):IJ=FN IN(J)
230 IF II<0 THEN TE=0:WHILE TE<200:SOUND
 1,TE,1:TE=TE+1:WEND:END
240 C=TEST (IB+8,IA-8)
250 IF C=1  THEN H=H+2*H*(IA>16*16 OR IA
<0):K=K+2*K*(IB>15*16 OR IB<0):SOUND 1,4
40,1:IF RND<0.5 THEN 280 ELSE GOTO 170
260 IF C=0 THEN MOVE IB,IA:PRINT CHR$(23
1);:T=1:GOTO 170
270 IF C=2 OR C=3 THEN MOVE IB,IA:PRINT
CHR$(231);:H=-H:K=K*(1+2*(RND<0.5)):T=1:
S=S+C:SOUND 1,200,6
280 K=SGN(K)*(16+10*(RND<0.5))
290 GOTO 170
```

Le lézard d'Alice

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

LE JEU :
Il s'inspire très vaguement d'une idée de LEWIS CARROLL, dans *Alice au Pays des Merveilles*. Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu, pour des raisons d'encombrement de la mémoire, c'est un lézard qui change de taille.
Deux verres apparaissent sur l'écran : l'un contient le liquide qui fait grandir, l'autre le liquide qui fait rapetisser (malheureusement ce n'est pas toujours le même verre). Le joueur devra indiquer la quantité de liquide qu'il désire boire et le lézard changera de taille en fonction de cette quantité. Le but du jeu est de parvenir à lui rendre sa taille initiale.

LE PROGRAMME :
La boucle I (5Ø-8Ø) dessine les verres.
Le sous-programme 43Ø est appelé si la réponse à la question QUANTITE n'est pas assez rapide. La ligne 22Ø annule cet appel.
La boucle I (23Ø-25Ø) les remplit avec la quantité indiquée en ligne 21Ø.
La variation de taille du lézard est définie par la ligne 29Ø, avec contrôle par les lignes 3ØØ-31Ø.
La boucle I (32Ø-34Ø) dessine le corps du lézard, les lignes 35Ø et 36Ø dessinent les pattes, la ligne 37Ø la tête ; les deux boucles séparées I (38Ø et 39Ø) forment le bec et les dents.
Un message gagnant sera affiché par la ligne 41Ø si l'on a trouvé la bonne taille.

POUR JOUER :
On lance le programme par RUN : les deux verres se dessinent, puis apparaît la question QUEL COTE à laquelle on répond en tapant G ou D ; on obtient l'affichage QUELLE QUANTITE ; on entre alors une quantité (de Ø à 2ØØ) et on frappe ENTER.
Le verre désigné se remplit sous vos yeux, puis, après quelques instants, le lézard est affiché. On devra alors frapper une touche quelconque ; si le lézard n'avait pas la bonne taille, aucun message n'est affiché et l'on revient au dessin des verres.

EXTENSIONS POSSIBLES :
Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins et la couleur, de modifier le caractère aléatoire du choix des verres, etc.
Pourquoi ne pas relire *Alice au Pays des Merveilles,* les idées n'y manquent pas.

```
10 REM LE LEZARD D'ALICE
20 RANDOMIZE(TIME)
30 MODE 0:N=33
40 PAPER 5
50 FOR I=1 TO 9
60 LOCATE 1,I+4:PRINT " ":LOCATE 8,I+4:P
RINT " "
70 LOCATE 13,I+4:PRINT " ":LOCATE 20,I+4
:PRINT " "
80 NEXT I
90 FOR I=1 TO 8:LOCATE I,14:PRINT " ":NE
XT I
100 FOR I=13 TO 20:LOCATE I,14:PRINT " "
:NEXT I
110 PAPER 0:PEN 3:LOCATE 2,15:PRINT "GAU
CHE":PEN 8:LOCATE 14,15:PRINT "DROIT"
120 PEN 1:LOCATE 1,17:PRINT "QUEL COTE :
"
130 A$=INKEY$:A$=UPPER$(A$):IF A$<>"D" A
ND A$<>"G" THEN 130
140 U=(1-12*(A$="D"))*32:C=3-5*(A$="D")
150 PEN C:LOCATE 12,17:IF A$="D" THEN PR
INT "DROIT" ELSE PRINT "GAUCHE"
160 PEN 1:PRINT:PRINT "QUANTITE  :";:PEN
 C:F=0:R$=""
170 AFTER 100,0 GOSUB 430
180 A$=INKEY$:IF F=1 THEN GOTO 260
190 IF A$="" THEN GOTO 180
200 IF A$<>CHR$(13) THEN R$=R$+A$:PRINT
A$;:GOTO 180
210 Q=ABS(VAL(R$))
220 Z=REMAIN(0)
230 FOR I=1 TO 8*SQR(Q)
240 MOVE U,12*16+I:DRAWR 6*32-2,0,C
250 NEXT I:SOUND 1,478:SOUND 1,426:SOUND
 1,379
260 TE=0:WHILE TE<600:TE=TE+1:WEND
270 CLS
280 H=1+2*(RND<0.5)
290 N=N+H*INT(SQR(Q))
300 IF N<12 THEN N=12
310 IF N>60 THEN N=60
320 FOR I=1 TO 10*N
```

```
330 MOVE I+32,195:DRAWR 0,10,12
340 NEXT I
350 TAG:MOVE 32,192:PRINT CHR$(153);
360 MOVER 10*N-64,0:PRINT CHR$(153);
370 MOVE 32,220:PRINT CHR$(224);:TAGOFF
380 FOR I=0 TO 1:MOVE 0,212+2*I:DRAWR 32
,I:NEXT I
390 FOR I=1 TO 4:MOVE 7*I,211:DRAWR 0,-2
:NEXT I
400 A$=INKEY$:IF A$="" THEN 400
410 IF N=28 THEN LOCATE 8,23:PRINT "BRAV
O!":SOUND 1,478:SOUND 1,379:SOUND 1,319:
END
420 CLS:GOTO 40
430 LOCATE 2,23:PEN 2:PRINT "TROP LONG,
JE JOUE"
440 Q=INT(RND*300+10):F=1
450 RETURN
```

Conseils de lecture

Pour approfondir vos connaissances en BASIC, et mieux connaître le système des CPC 464, 664 et 6128, ainsi que du PCW 8256, P.S.I. vous propose une palette d'ouvrages utiles.

POUR JOUER AVEC OU CONTRE VOTRE AMSTRAD

- **Super-jeux Amstrad,** par Jean-François Sehan (Editions du P.S.I.) : 50 programmes BASIC sont ici abondamment commentés et totalement ouverts à toute modification et adaptation. Le niveau de programmation est supérieur à celui du "102 programmes".

- **Amstrad en famille,** par Jean-François Sehan (Editions du P.S.I.) : Pour gérer votre compte en banque, votre portefeuille boursier ou votre congélateur, pour apprendre ou jouer en famille avec votre micro, cet ouvrage vous propose des programmes BASIC faciles à rentrer et adaptables à volonté.

POUR MAITRISER LE BASIC AMSTRAD

- **Découverte de l'Amstrad,** par Daniel-Jean David (Editions du P.S.I.) : Une découverte exhaustive et claire des CPC 464, 664 et 6128, un apprentissage progressif du BASIC Amstrad pour débutants complets.

- **BASIC Amstrad – 1. Méthodes pratiques,** par Jacques Boisgontier et Bruno Césard (Editions du P.S.I.) :
  Pour ceux qui connaissent déjà un BASIC, voici un bon ouvrage de perfectionnement à la programmation en BASIC sur Amstrad 464, 664 et 6128.

- **BASIC Amstrad – 2. Programmes et fichiers,** par Jacques Boisgontier (Editions du P.S.I.) :
  Pour programmer le BASIC AMSTRAD, cet ouvrage donne de nombreux programmes de gestion, d'éducation et de jeux où le rôle des fichiers est expliqué et largement commenté.

- **BASIC plus, 80 routines sur Amstrad,** par Michel Martin (Editions du P.S.I.) :
  Pour pousser votre Amstrad au maximum de ses capacités : 80 routines de simulation d'instructions qui n'existent pas en BASIC Amstrad.

- **Périphériques et gestion de fichiers sur Amstrad,** par Daniel-Jean David (Editions du P.S.I.) :
  Cet ouvrage s'adresse aux lecteurs, déjà initiés à l'Amstrad, souhaitant programmer en BASIC Amstrad des applications utilisant des fichiers sur disquette ou cassette, et programmer tous les périphériques des CPC 464, 664 et 6128.

POUR MIEUX CONNAITRE LE SYSTEME DES CPC ET DU PCW 8256

- **Clefs pour Amstrad – 1. Système de base,** par Daniel Martin (Editions du P.S.I.) :
Mémento présentant synthétiquement le jeu d'instructions du Z80, les points d'entrée des routines système, les connecteurs et brochages, etc. Le livre de chevet du programmeur sur Amstrad.

- **Clefs pour Amstrad – 2. Système disque,** par Daniel Martin et Philippe Jadoul (Editions du P.S.I.) :
Ce deuxième tome consacré au système disque présente les points d'entrée des routines disque, les blocs de contrôle, la programmation et les brochages des circuits spécialisés... La deuxième partie du livre est aussi destinée aux possesseurs d'Amstrad 8256.

- **CP/M plus sur Amstrad,** par Yvon Dargery (Editions du P.S.I.) :
Toutes les commandes CP/M et CP/M plus pour maîtriser le système des 6128 et 8256 : un ouvrage de référence illustré par de nombreux programmes.

- **Le livre de l'Amstrad, tome 1,** par Daniel Martin et Philippe Jadoul (BCM - diffusé par P.S.I.) :
Ce livre destiné aux programmeurs des CPC 464 et 664, donne une étude complète de tous les circuits internes et analyse la structure interne du BASIC. Vous y trouverez, en outre, une étude complète des RSX, et des programmes de scrolling, de traçage de rectangles, de coloriage de surface et de manipulation vectorielle.

POUR ETRE INFORME REGULIEREMENT DE L'ACTUALITE DES MICROS AMSTRAD

- **MICROSTRAD,** revue bimestrielle du Groupe Tests.
Pour exploiter au mieux les capacités de votre micro, vous trouverez au sommaire de chaque numéro, un rendez-vous avec les rubriques clés :
  - Découvrez la face cachée de votre CPC : astuces, idées, conseils, tout pour comprendre votre micro, son anatomie, son fonctionnement, sa programmation et exploiter ses capacités graphiques et sonores.
  - Domptez votre CPC 464, 664, 6128 ou PCW 8256 : passionnés, petits ou grands, spécialistes ou débutants, une information pratique et la compétence d'experts au service de votre micro.
  - Programmez votre micro Amstrad : dans chaque numéro de MICROSTAD, un cocktail de programmes (dessins, jeux, utilitaires, gestion, etc.) et des trucs de programmation.

Achevé d'imprimer par l'imprimerie Tardy Quercy S.A. à Bourges
Dépôt légal : Février 1986 - N° d'Imprimeur : 12743
N° d'Édition : 86595-222-5

Votre avis nous intéresse

- Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent livre.
- Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous adresserons votre demande à l'auteur qui ne manquera pas de vous répondre directement.

- Ce livre vous donne-t-il toute satisfaction?

- Y a-t-il un aspect du problème que vous auriez aimé voir abordé?

Comment avez-vous eu connaissance de ce livre?

- ☐ publicité
- ☐ catalogue
- ☐ boutique micro
- ☐ autres
- ☐ cadeau
- ☐ librairie
- ☐ exposition

Avez-vous déjà acquis des livres PSI?

lesquels? ________

qu'en pensez-vous? ________

Nom ________ Prénom ________ Age ____

Adresse ________

Profession ________

Centre d'intérêt ________

CATALOGUE GRATUIT

Vous pouvez obtenir un catalogue complet des ouvrages PSI, sur simple demande, ou en retournant cette page remplie à votre libraire, à votre boutique micro ou aux

Editions du PSI
BP 86
77402 Lagny-sur-Marne Cedex

Apprendre en se distrayant, tel est l'objectif de ce livre.
Ces 102 programmes de jeu vous guideront progressivement dans l'exploration du BASIC **Amstrad**.
Niveau par niveau, vous assimilerez de nouvelles instructions pour arriver à la maîtrise de votre **CPC 464**.
Tous les programmes contenus dans ce livre fonctionnent également sur **CPC 664 et 6128**.
Chaque niveau commence par une présentation pédagogique du jeu d'instructions utilisé.
Tous les jeux sont décrits et les programmes sont analysés ligne par ligne afin d'en faciliter la modification.
Devenez vite le chef d'orchestre de votre **Amstrad** !

DES LIVRES POUR VOTRE ORDINATEUR

ISBN : 2.86595.222.3
120 FF
9 782865 952229

Couverture GIBRAT

102 PROGRAMMES POUR AMSTRAD

PAR JACQUES DECONCHAT

PROGRAMMES

P.S.I.